

Hitachi *Virtual Storage Platform* Enterprise Storage System

– Architecture and Concepts

A White Paper

By Alan Benway (Performance Measurement Group, Technical Operations)

Hitachi Data Systems Partner and Internal Use Only

May 22, 2013



Notices and Disclaimer

Copyright © 2013 Hitachi Data Systems Corporation. All rights reserved.

The performance data contained herein was obtained in a controlled isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While Hitachi Data Systems Corporation has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results can be obtained elsewhere.

All designs, specifications, statements, information and recommendations (collectively, "designs") in this manual are presented "AS IS," with all faults. Hitachi Data Systems Corporation and its suppliers disclaim all warranties, including without limitation, the warranty of merchantability, fitness for a particular purpose and non-infringement or arising from a course of dealing, usage or trade practice. In no event shall Hitachi Data Systems Corporation or its suppliers be liable for any indirect, special, consequential or incidental damages, including without limitation, lost profit or loss or damage to data arising out of the use or inability to use the designs, even if Hitachi Data Systems Corporation or its suppliers have been advised of the possibility of such damages.

Virtual Storage Platform® and **Universal Storage Platform® V** are registered trademarks of Hitachi Data Systems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

This document has been reviewed for accuracy as of the date of initial publication. Hitachi Data Systems Corporation may make improvements and/or changes in product and/or programs at any time without notice.

No part of this document may be reproduced or transmitted without written approval from Hitachi Data Systems Corporation.

WARNING: This document can only be used as HDS internal documentation for informational purposes only. This documentation is not meant to be disclosed to customers or discussed without a proper non-disclosure agreement (NDA).

Document Revision Level

<i>Revision</i>	<i>Date</i>	<i>Description</i>
1.0	October 2010	Initial Release
2.0	October 2011	Major revision
2.1	July 2012	Cache and cache management changes, various updates
2.2	August 2012	Update PCIe specifications, discussion
2.3	September 2012	Fix typos, add some extra PCIe information
2.4	May 2013	FBX and FCoE Updates, wording changes to match HUS VM paper, code update related changes

References

The following link is the internal access point for all papers (and some presentations) produced by the Performance Group in TechOps (document names will change over time).

<http://sharepoint.hds.com/sites/techopsSRC/Performance%20White%20Papers/Forms/AllItems.aspx>

Hitachi USP V Architecture & Concepts Guide June 2009

Hitachi VSP HDP and HDT Architecture, Concepts, and Best Practices September 2011

Hitachi VSP Cache Operations Concepts (coming June 2013)

Hitachi VSP Architecture Deep Dive presentation

Contributors

The author would like to recognize and thank the following reviewers of the 2.4 version of this document:

Wendy Roberts, Technical Director, Enterprise Storage, APAC

Ian Vogelesang, Performance Measurement Group - Technical Operations, HDS

Rob Whalley, Product Manager, Enterprise Storage Systems, EMEA

Table of Contents

Introduction	8
Overview of Changes	8
VSP: A Unique 3D Storage system Design.....	10
Scale Up – Single Control Chassis	10
Scale Out – Dual Control Chassis.....	10
Scale Deep – Virtualization	11
Hitachi Dynamic Provisioning	11
Hitachi Dynamic Tiering.....	12
Glossary	13
Basic Theory of Operation.....	17
Architecture Overview	18
Frames	20
DKUs and HDUs.....	22
FBXs and FMUs	23
Logic Box and Logic Boards.....	24
Array Limits.....	26
Logic Board Details.....	27
HiStar-E Grid Switch (GSW).....	27
Overview	27
Virtual Storage Director Board (VSD).....	28
Overview	28
LDEV Ownership by VSD (MPU)	30
VSD Command Tags	31
Data Cache Adapters	31
Overview	31
SSD Drives for Backup	33
Data Cache	33
Control Memory.....	33
Front end Director Boards (FEDs).....	34
Overview	34
Open Fibre 16-Port Feature	34
FCoE 8-port Feature	35



FICON 16-port Feature	36
Back end Director Boards (BEDs)	37
Overview	37
Data Accelerator	38
SPC Processor.....	39
SAS Engine Concepts.....	39
BED SAS Links to FBX/DKU HDU Overview	39
FBX Details.....	42
DKU Details	42
DKU- HDU Internal Details.....	47
FBX- HDU Internal Details.....	48
Disk Details	48
Installable Drive Features.....	48
FMD Drives.....	49
SSD Drives	50
SAS Drives	50
SATA Drives	51
SATA Drive Sector Details	52
Cache and RAM Overview	53
Memory Organization	53
Cache System.....	53
VSD Board Local Memory.....	54
DXBF: FED and BED Board Local Memory	54
FED Open Fibre Port Operations	54
I/O Request Limits and Queue Depths.....	54
External Storage Mode I/O (Open Fibre FED)	55
Cache Mode Settings.....	55
Other External Mode Effects	55
VSP Chassis Bandwidth Overview	56
General Storage Concepts	58
Understand Your Customer's Environment.....	58
RAID Levels and Write Penalties	58
Parity Groups and Array Groups	59



RAID Chunks and Stripes.....	60
LUNS (host volumes)	60
Number of LUNs per Parity Group	61
Port I/O Request Limits, LUN Queue Depths, and Transfer sizes	61
Port I/O Request Limits	61
Port I/O Request Maximum Transfer Size.....	61
LUN Queue Depth.....	62
Mixing Data on the Physical drives.....	62
Workload Characteristics.....	62
Selecting the Proper drive drive Form Factor	62
Mixing I/O Profiles on the Physical drives	63
Front end Port Performance and Usage Considerations.....	63
Host Fan-in and Fan-out	63
Mixing I/O Profiles on a Port.....	64
Summary	64
Appendix 1. Open Systems RAID Mechanisms	66
Appendix 2. Mainframe 3390x and Open-x RAID Mechanisms	68
Appendix 3. Putting SATA drive Performance into Perspective	71
Appendix 4. LDEVs, LUNs, VDEVs and More.....	75
Logical Devices (VSP, USPV)	75
HDEVs and LUNs or LUSE	77
VDEVs	79
Internal VDEV	80
External VDEV	80
CoW VDEV	81
Dynamic Provisioning VDEV	81
Layout of internal VDEVs on the parity group	82
Appendix 5. Concatenated Parity Groups.....	86
Advantages of concatenated Parity Groups.....	88
Disadvantages of concatenated Parity Groups:	89
Using LDEVs on concatenated Parity Groups as Dynamic Provisioning pool volumes	89
Summary of the characteristics of concatenated Parity Groups (VDEV interleave):	90
Appendix 6. VSP B4 Maps – Parity Groups Names	91
Appendix 7. USP V Architecture Overview	92



Appendix 8. VSP Hardware Maintenance Manual, page THEORY02-210 Explained.....	93
Appendix 9. VSP Open System FED Port Names	95
Appendix 10. PCI Express Details.....	97
Appendix 11. VSP LDEV Ownership and Performance.....	99
Performance Advantages	99
Potential Performance Drawbacks	99
Cache Allocation per LDEV	100
DPVol Considerations	100
CLPR Usage Considerations	100
Summary.....	101

Hitachi *Virtual Storage Platform* Enterprise Storage System

– Architecture and Concepts

A White Paper

By Alan Benway (Performance Measurement Group, Technical Operations)

Introduction

This document covers the hardware architecture and functional concepts of that hardware for the **Hitachi Virtual Storage Platform (VSP)** enterprise storage system. The physical hardware and operational details are covered in the following sections. This document is not intended to cover any aspects of program products, customer application software, customer specific environments, or features available in future releases.

This document is intended to familiarize Hitachi Data Systems' sales personnel, technical support staff, approved customers, and value-added resellers with the features and concepts of the VSP. The users that will benefit the most from this document are those who already possess an in depth knowledge of the previous **Hitachi Universal Storage Platform V (USP V)** architecture.

This document will receive future updates to refine or expand on some discussion as the internals of the design are better understood.

Overview of Changes

Replacing the proven and superior Hitachi USP V technology, the Hitachi VSP offers an entirely new level of scalable enterprise storage, capable of meeting the most demanding of workloads while providing great flexibility. The VSP offers much higher performance, better scalability, higher reliability, and greater flexibility than any competitive offering today.

These are the new features that distinguish the VSP from the previous USP V model:

- The VSP can be configured as a storage system with a *single control chassis* (logic box) with 1 to 3 frames, or as a *dual control chassis* storage system, each with 1-3 frames (or 6 overall). Note that a frame is a rack. The dual chassis system is still a unified or single storage system; it is scaled up.
- The USP V's Universal Star Network crossbar switch has been replaced by the **HiStar-E PCI Express Switched Grid**. See Appendix 10 for details on PCIe connections.
- The USP V used four dedicated NEC processors (the "MP") on each Front End Director (FED) and Back End Director (BED) board that managed all I/O operations. That has been replaced by a design that uses shared Virtual Storage Director boards (**VSDs**) with quad-core Xeon processors

(**MPUs**) that manage all I/O according to sets of assigned LDEVs (LUNs, as seen by the hosts). The VSD Intel CPUs work in concert with the multifunction **Data Accelerator** processors on the all-new FEDs and BEDs.

- The 800MHz NEC CPUs and their support chips (used on USP V FED and BED boards) have been replaced by the **2.33GHz Intel Xeon Core Duo** quad-core CPUs and their support chips on the VSD boards.
- The individual USP V BED boards provided four 4Gbps FC-AL loops to the drive containers, whereas the **VSP BED** boards each have dual SAS controllers providing 8 x 6Gbps full duplex SAS links to as many as 64 SAS switches within the drive containers. These switches are inside the Flash FMU trays (FMD drives), or the Large Form Factor (3.5") or Small Form Factor (2.5") drive trays (HDUs within DKUs) that hold SSDs, SAS or SATA drives.
- The USP V's **Shared Memory (SM)** system uses eight dedicated SM paths on every FED and BED board which are attached to the four dedicated Shared Memory boards with dedicated RAM for all system metadata and control tables. This design has been replaced by an all-new **Control Memory (CM)** design. Control Memory metadata and control tables now reside in high speed local RAM on each VSD processor board, plus a master copy in a reserved section on the first two boards of system cache. The majority (perhaps 80%) of all Control Memory accesses are reads to the local VSD board DRAM copy. Updates are written to both the VSD's local DRAM and the master CM copy in cache.
- Each **VSD** board's single **MPU** controls all I/O operations for a discrete group of LDEVs (LUNs when they are mapped to a host port). As they are configured, LDEVs are automatically assigned in a round-robin fashion to one of the MPUs. It is possible to manually reassign individual LDEVs to another MPU (VSD board) to manually rebalance the loads across the installed VSDs. Note that this can be a very lengthy process. Also, LDEVs can be automatically redistributed across all MPUs when adding more VSD boards to a system (a storage system mode setting controls this behavior).

Each MPU executes the code for the following processes:

- o target mode (Mainframe or Open Systems hosts)
- o external mode (virtualization of other storage)
- o BED mode
- o MCU - Initiator for HUR or TC
- o RCU - Target for HUR or TC

Instead of this code being a fixed microcode load per individual FED/BED board MP processor - as on the USP V - it is now simply a process scheduled on a per job basis within one of the four Intel processor cores per MPU. The MPUs control the FED and BED boards, maintain Control Memory and manage all cache usage.

- An installed VSP may be scaled up from a single controller chassis (DKC-0) to a dual controller-chassis system (DKC-0 and DKC-1) where each chassis has their own control frame, logic box and controller boards. The dual chassis VSP still runs a single system image. Each controller chassis can have up to three frames with up to eight drive containers (**DKUs**) attached per controller chassis. There is a 2.5" form factor DKU (the **SBX**, with 128 drives) and a 3.5" form factor DKU (the **UBX**, with 80 drives) to choose from. Some drives are 2.5" (Small Form Factor, or SFF) while others are 3.5" (Large Form Factor, or LFF). Two of these DKU boxes could instead be FBX boxes (Accelerated Flash feature) with up to 48 Hitachi FMD drives installed in four FMUs.

- Current drive choices include SFF 200GB and 400GB MLC SSD drives; SFF 146GB 15K, 300GB 15K, 300GB 10k, 600GB 10K, and 900GB 10k SAS drives; LFF 2TB SATA and 3TB 7.2K SAS drives; 1.6TB FMD drives.
- The VSP's upper limit on drive capacity in a dual chassis configuration is 1280 (3.5" LFF) or 2048 (2.5" SFF), or some intermediate value in between (if using mixed DKU container types in a system). This includes up to 192 of the new FMD Accelerated Flash drives.
- There is a software feature that is layered above the **Hitachi Dynamic Provisioning** feature called **Hitachi Dynamic Tiering**. This allows for the use of different drives types and RAID levels within a single virtual HDP Pool (a wide-striped mechanism), with the physically allocated 42MB Pool pages (38MB on Mainframe) automatically migrating among the storage tiers according to frequency of use by the associated Dynamic Provisioning Volumes (DPVol, what is seen by the host).

VSP: A Unique 3D Storage system Design

The Virtual Storage Platform is an all-new Hitachi enterprise storage system that supports a large number of intermixed SSD, SAS, and SATA drives. It is best described as a **3D storage system**, due to the following characteristics within the system (without any external in-band or out-of-band devices required to manage storage):

- A VSP solution scales **Up** (hardware within a single control chassis storage system) for increased performance and connectivity
- A VSP solution scales **Out** (upgrade to a dual control chassis storage system) for doubled internal capacity, connectivity, and performance
- A VSP solution scales **Deep** with virtualization of external storage systems, with the use of additional tier-2 and tier-3 performance and capacity

Scale Up – Single Control Chassis

Within the basic, single control chassis storage system, the number of logic boards, drive containers, and drives is highly scalable. A system can start with the minimum set of 10 logic boards in one control unit chassis and 1 disk drive chassis containing up to 80 LFF, 128 SFF, or 48 FMD drives. A diskless system can start with the minimum set of 8 logic boards and no internal storage and only use external virtualized storage attached to FED ports. The system can then be expanded with more logic boards (up to 28 in all) and more drive containers (up to 8 in all) and/or more external storage. A single control chassis can have a maximum of two FMD disk drives chassis and six SFF/LFF disk chassis, or eight SFF/LFF disk chassis in any combination. In a pure SFF disk chassis configuration, a single storage system can have up to 1024 drives in 8 SFF disk chassis. If using only LFF chassis, the limit is 640 drives in 8 LFF disk chassis. There can be total of 96 FMD drives in two FBX disk chassis.

Scale Out – Dual Control Chassis

A VSP's resources can be greatly extended (everything doubled) by the use of a dual controller storage system (still a single system, not a cluster) with up to six frames in all. A dual controller storage system can start with a minimum set of 10 logic boards and 1 disk drive chassis containing up to 80 LFF, 128 SFF, or 48 FMD drives. A diskless system can start with the minimum set of 8 logic boards and no internal storage and only use external virtualized storage attached to FED ports.

In a dual controller storage system, a minimum of two frames is required each containing one control unit chassis which contain the logic boards for a maximum of 48 logic boards. These two frames can also support up to an additional 4 disk chassis (2 per frame), with each disk chassis containing up to 80 LFF, 128 SFF, or 48 FMD drives. Four additional storage-only frames, a maximum of 2 on the right side and 2 on the left side, can hold up to 3 disk chassis each containing up to 80 LFF, 128 SFF, or 48 FMD drives. A maximum of 192 FMD drives are supported in a 4x FBX disk chassis and 12x disk drive chassis using SFF/LFF disk drive chassis in any combination. A maximum of 16 disk chassis of SFF/LFF disk drives is supported in any combination.

The logic box in each control frame is the same, using the same types and numbers of logic boards. The two logic boxes are cross-attached at the internal grid switch level creating a global cache system. The names of the boards and FED ports in the second control chassis are changed from what they would be in a single chassis storage system.

Note that any front end port on a dual chassis storage system can access any internal (or mapped external) LUN as there is no host-visible division within the storage system. The only hardware division is that each DKC has its own BEDs and associated FBXs and DKUs. But any MPU processor can manage a back-end I/O operation on any BED board in the system. In a pure SFF configuration, a dual chassis storage system can have up to 2048 drives in 16 DKUs, or if using only LFF drives, the limit is 1280 drives in 16 DKUs. It can also have up to 192 FMD drives (replaces four DKUs with four FBXs).

Scale Deep – Virtualization

The VSP provides an enhanced version of the Hitachi Virtualization mechanism (the Universal Volume Manager feature) as found on the USP V/VM. Existing customer storage systems (usually repurposed upon being replaced by newer storage systems) may be attached to some of a VSP's FED FC ports (now operated in *External Initiator Mode* rather than host *Target* mode). The LUNs that are supplied by these external storage systems are accessed and managed via hosts that are attached to other FED FC, FCoE, or FICON ports (operated in Target Mode). As far as any host is concerned, all virtualized LUNs passed through the VSP external ports to the host target ports simply appear to be other internal LUNs from the VSP. The VSP's FC FED ports that attach to the external storage system's front end ports are operated in "external" or "SCSI initiator" mode (attached to storage), rather than the usual "SCSI target" mode (attached to hosts).

Virtualized (external) storage should be used as Tier-2 and Tier-3, not Tier-0 or Tier-1, which should be limited to the internal VSP Parity Groups that use SSD or SAS drives (but not SATA). One benefit of virtualization is greatly simplified management of LUNs and hosts. Another major advantage is the ability to dynamically and transparently relocate LUNs from Tier-1 RAID Groups down to Tier-2 or Tier-3 using other RAID levels, drive types, or external storage. These migrations are able to proceed while the original source remains on line to the hosts, and the VSP will seamlessly switch over the mapping from the original LUN to the new LUN when completed. No changes on the host mount point are required.

Hitachi Dynamic Provisioning

Hitachi Dynamic Provisioning (HDP) was introduced on the USP V/VM storage systems. It provides a logical mechanism for grouping multiple RAID Groups into a single Pool (with many such Pools possible on one system) that presents a wide striped block device to a host. A Pool may be created for either for Mainframes or Open Systems. A standard (not HDT) HDP Pool is defined with two or more Parity Groups (Pool Volumes) with one RAID level and one drive type. Use of 32 to 128 drives per Pool would be more typical. The Open Systems Pool mechanism creates a structure of 42MB Pool pages from each RAID

Group across the entire space when initially created. The Mainframe Pool mechanism uses a 38MB page size. This is similar (at a basic level) to the use of a host-based logical volume manager (LVM) with its wide striping across all member LUNS in its pool with use of a (typically) large volume chunk size (usually 1MB or greater).

Dynamic Provisioning Volumes (DPVOLs or virtual volumes) are then created with a user specified *logical* size (up to 60TB) and emulation mode (Open Systems or Mainframe types), then associated with a single Pool. The host accesses the DPVOL (or many of them – even hundreds) as if it were a normal volume (LUN) over one or more host ports. A major difference is that physical space is not allocated to a DPVOL from the Pool's physical space until the host writes to different parts of that DPVOL's Logical Volume Address (**LBA**) space. The entire logical size specified when creating that DPVOL could eventually become fully mapped to physical space using **38MB** or **42MB Pool pages** from every RAID Group in the Pool. If new RAID Groups are added to a Pool later on, a rebalance operation of the currently allocated Pool pages onto these additional Parity Groups can be initiated.

Hitachi Dynamic Tiering

Hitachi Dynamic Tiering (**HDT**) is a feature that enhances the operation of an HDP Pool. Previously, each HDP Pool had to be created using one RAID level and one drive type. For example, a *high performance* Tier-1 Pool could be set up using 15K RPM SAS drives in a RAID-10 (2D+2D) configuration. A *standard performance* Tier-2 Pool could use 10K RPM SAS drives and RAID-5 (such as 3D+1P). An *archive performance* Tier-3 Pool could be set up with 7200 RPM SAS drives in RAID-5 or RAID-6 for near-line storage. Tier-2 and Tier-3 Pools could also be created using LUNs from virtualized storage (typically with SAS drives for Tier-2 and SATA drives for Tier-3).

The new HDT feature allows a *single* Pool to contain multiple types of RAID group (Pool Volumes) and any type of drive, as well as external LUNs from virtualized storage. Up to three choices from these possible combinations are allowed per Pool. An example would be SSDs (Tier-1), SAS 10K (Tier-2), and external storage using SAS 7.2k (Tier-3). Only one RAID level is normally used per Tier. However, when a Tier is to be changed from a drive type or RAID level to another, that one Tier may contain two types (drive, RAID level) for the time the migration is in progress. For example a Tier-2 may have been established using SAS 15K drives and RAID-5 (7D+1P), but it is desired to change this to SAS 10K and RAID-6 (6D+2P). Both types of Pool Volumes will be in that Tier until the migration is complete and the original Pool Volumes removed. The original pool volume may be deleted using **Pool Shrink**, and in doing so the HDT software will relocate all allocated 42MB pages from that pool volume to all the new ones. Once the copy is completed (may take a long time) then that pool volume (an LDEV) is removed from that Pool. That LDEV can now be reused for something else (or deleted, and the drives for that Parity Group removed from the system).

HDT manages the mapping of 42MB Pool pages (38MB for Mainframe Pools) within these various tiers within a Pool automatically. Management includes the dynamic relocation of a page based on frequency of back end disk I/O to that page. Therefore, the location of a Pool page (42MB or 38MB) is managed by HDT according to host usage of that part of an individual DPVOL's LBA space. This feature can eliminate most user management of storage tiers within a subsystem and can maintain peak performance under dynamic conditions without user intervention. This mechanism functions effectively without visibility into any file system residing on the volume. The top tier is kept full at all times, with pages being moved down a tier to make room for new high-activity pages. This is a long term space management mechanism, not a real-time relocation service.

Glossary

At this point some definitions of the various terminology used is necessary in order to make all of the following discussions easier to follow. Throughout this paper the terminology used by Hitachi Data Systems (not Hitachi LTD.) will normally be used. As a lot of storage terminology is used differently in Hitachi documentation or by the users in the field, here are the definitions as used in this paper:

- **Accelerated Flash** (installable) – a new type of storage device available on a VSP. This option is based on the use of the 8U **FBX** drive chassis (similar to the function of the DKU), an **FMU** 12-slot tray (four per FBX), and an **FMD** device (12 per FMU).
- **Array Group** (installable, drive feature) – The term used to describe a set of four physical drives installed as a set into four HDU containers (on a horizontal slice) within a single DKU or FBX box. When a set of one or two such Array Groups (four or eight drives) is formatted using a RAID level, the resulting *RAID formatted* entity is called a **Parity Group**. Although technically the term Array Group refers to a group of bare physical drives, and the term Parity Group refers to something that has been formatted as a RAID level and therefore actually has parity data (here we consider a RAID-10 mirror copy as parity data), be aware that this technical distinction is often lost and you will see the terms Parity Group and Array Group used interchangeably by the field.
- **BED** (installable, **Back End Director** feature) - the VSP drive controller board that provides four central Grid Switch links, SAS controllers, and eight back end 6Gbps SAS links via four SAS 2-Wide ports. A pair of boards (thus 16 SAS links) is always installed as a feature. (DKA board)
- **Cache Directory** – the 500MB region reserved on each VSP Data Cache Adapter (cache board) for use in managing that DCA's cache slot allocations to the installed VSDs or the global free list.
- **CHA** (Channel Adapter) - Hitachi's name for the **FED** board.
- **Chassis** (a CBXA or CBXB) – the 14U **logic box** in the controller frame (DKC) which holds up to 28 boards. In the first controller frame (DKC-0) it is called the **CBXA**, while the one in the second controller frame (DKC-1) it is the **CBXB**. A **single chassis storage system** means there is one control frame with a logic box. A **dual chassis storage system** means that there are two control frames, each with a logic box, and these two logic boxes are tied together via 8 or 16 external Grid Switch link cables.
- **CMA** (Cache Memory Adapter) - Hitachi's name for the DCA board.
- **Concatenated Parity Group** - A configuration where the VDEVs corresponding to a pair of RAID-10 (2D+2D) or RAID-5 (7D+1P) Parity Groups, or four RAID-5 (7D+1P) Parity Groups, are interleaved on a RAID stripe level on a round robin basis. A logical RAID stripe row is created as a concatenation of the individual RAID stripe rows. This has the effect of dispersing I/O activity over twice or four times the number of drives, but it does not change the number, names, or size of VDEVs, and hence it doesn't make it possible to assign larger LDEVs to them. Note that we often refer to RAID-10 (4D+4D), but this is actually two RAID-10 (2D+2D) Parity Groups interleaved together. For a more comprehensive explanation see the *Concatenated Parity Group* section in the appendix.
- **Control Memory (CM)** - the region in each MPU's local DRAM and in the global cache (on the first pair of DCAs) that is used to manage LDEV and DPVOL metadata and all system states. It replaces the USP V's use of a separate Shared Memory boards. In general, Control Memory contains all of the metadata in a storage system that is used to describe the physical

configuration, track the state of all LUN data, track the status of all components, and manage all control tables that are used for I/O operations (including those of Copy Products). The overall size of Control Memory in cache can grow up to **56GB** (28GB + 28GB mirror).

- **Controller Frame** – the VSP frame (rack) that contains the logic box and up to 28 logic boards. A dual chassis storage system has two of these frames. In addition to the logic box at the bottom of the frame, the control frame also has room for 1 or 2 optional DKU or FBX boxes located at the middle and upper portions of the frame.
- **DCA** – (installable, **Data Cache Adapter**) the VSP cache board, with 4 PCIe 1.x x4 Grid Switch links, 8 DIMM slots in 4 banks of DRAM for DDR3-800 DIMMs, and one or two SSDs for local (on-board) cache backup.
- **DKC** – **DisK Controller** Hitachi's name for either a logic box (i.e. DKC-0, DKC-1) or the controller frame.
- **DKA** – (**DisK Adapter**) Hitachi's name for the BED.
- **DKU** (installable, **DisK Unit**) – the 13U **SBX** 128 drive (2.5" devices) or 13U **UBX** 80 drive (3.5" devices) container (also known as a drive *chassis*) that mounts in a frame. There can be two DKUs in a control frame (the DKC) and three in each expansion (DKU) frame. Each DKU contains eight **HDU** boxes. Each DKU is also split vertically across two power domains, with four HDUs in each power domain.
- **DPVOL** – (configurable, **Dynamic Provisioning VOLUME**) - the Virtual Volume assigned to an HDP Pool. Some documents also refer to this as a **V-VOL**. It is a member of a V-VOL Group, which is a kind of **VDEV**. Each Open DPVOL has a user specified size between 8GB and 60TB in increments of one block (512 byte sector) and is built upon a set of 42MB pages of physical storage (so each DPVOL should be specified as a multiple of 42MB). On a Mainframe DPVOL the page size is reduced to 38MB.
- **Drive (disk)** – an FMD drive, SSD drive, a SAS drive, or a SATA drive.
- **DRR - Data Recovery and Reconstruction** processors that manage RAID parity operations, data encryption, and drive formatting or rebuilds.
- **eLUN** (configurable, **external LUN**) – an external LUN is one which is located in another storage system that is attached via two or more **ePorts** on a VSP and accessed by a host through other VSP FED ports (FC, FCoE, or FICON).
- **ePort** (configurable, **External Port**) – A FED port that is operated as an *external* port rather than a *target* port, an MCU port, or an RCU port. An external storage system directly connects to a VSP via two or more Fibre Channel FED ports instead of to a host. Those FED ports used in this manner are changed from the default *host target* mode to an *initiator* port (or external Port) mode by use of the **Universal Volume Manager** (UVM) software product. The VSP will 'discover' any external storage system exported LUNs on each ePort. The eLUN is used within the VSP as a **VDEV**, a logical container from which LDEVs can be carved. Individual external LDEVs may be mapped to a portion of or to the entirety of the eLUN. Usually a single external LDEV is mapped to the exact LBA range of the eLUN, and thus the eLUN can be "passed through" the VSP to the host.
- **FBX** – the rack mount box that contains four FMU 12-slot FMD drive boxes

- **Feature** (package) – an installable hardware option (such as a FED, BED, CDA, VSD, GSW, DIMMs or drives) that includes either two PCBs (one located on each VSP power domain or “cluster”) or four DIMMs or four drives.
- **FED** – (installable, **Front End Director**) - the Open Fibre Channel, Fibre Channel over Ethernet, or FICON interface board used to attach hosts to the storage system. Each has four Grid Switch links and eight 8Gbps host ports. The Open FC boards may also be used to attach local external storage, or to remote storage systems using the Hitachi Universal Replicator product.
- **FMD** – the accelerated flash drive module (currently 1.6TB)
- **FMU** – the 12-slot drive box for FMD modules , organized as two 6-slot HDUs
- **Frame** – preferred name for “rack”
- **GSW** – (installable, **Grid Switch**) the HiStar-E board (24 ports, PCIe 1.x x4 links, full duplex 1GB/s send and 1GB/s receive) which interconnects the FEDs, BEDs, DCAs, and VSD boards. (ESW)
- **HDU (Hard Disk Unit)** - the drive container within the FBX or DKU box. There are eight HDUs per FBX/DKU. On a DKU, half of the HDUs are presented on the front side and half on the rear side. Each FMD HDU holds 6 drives, each SFF (2.5”) HDU holds 16 drives, and each LFF (3.5”) HDU contains 10 drives. An entire DKU will either be SFF or LFF, but the two types of DKU boxes may be intermixed in a frame.
- **HiStar-E** – the name of the new PCI Express (PCIe 1.0) x4 link Grid Switch (GSW) interconnection path among the FED, BED, DCA, and VSD boards. Each such path (link) is operated as full duplex, with a simultaneous peak PCIe protocol rate of 1GB/s in each direction.
- **LDEV** (configurable, **Logical DEvice**) – A logical volume internal to the storage system that can be used to contain customer data. LDEVs are uniquely identified within the storage system using a six hex digit identifier in the form LDKC:CU:LDEV. LDEVs are carved from a VDEV (see VDEV), and thus there are four types of LDEVs - internal LDEVs, external LDEVs, COW V-VOLs, and DPVOLs. LDEVs may be mapped to a host as a LUN, either as a single LDEV, or as a set of up to 36 LDEVs combined in the form of a LUSE. Note: what is called an LDEV in HDS enterprise storage systems like the VSP is called an LU or LUN in HDS modular storage systems like the Hitachi Universal Storage (HUS) or Adaptable Modular Storage (AMS) family.
- **LUN** (configurable, **Logical Unit Number**) – the host-visible identifier assigned by the administrator to an existing LDEV to make it usable on a host port. An *internal* LUN has no actual queue depth limit (but 32 is a good rule of thumb) while an *external* (virtualized) eLUN has a Queue Depth limit of 2-128 (adjustable) per eLUN per external port path to that eLUN. In Fibre Channel, the host HBA Fibre Channel port is the *initiator*, and the storage system’s virtual Fibre Channel port or Host Storage Domain is the *target*. The Logical Unit Number is the *number* of the logical volume within a target.
- **LUSE** (configurable, **Logical Unit Size Expansion**) – A *concatenation* (“fill-and-spill”) of 2 to 36 LDEVs (up to a 60TB limit) that is then presented to a host as a single LUN. A LUSE will normally perform at the level of just one of these LDEVs (Parity Group). You may also see the term “HDEV” or Head LDEV. A LUSE is identified by the LDEV name of the first or “head” LDEV in the LUSE. The HDEV is the object that can be mapped to the host as a LUN, and it is either a single LDEV or a set of 2 to 36 LDEVs comprising a LUSE. LDEVs that are interior to a LUSE (not the head LDEV) may not be mapped to the host as a LUN. HDEVs may have multiple host paths,

meaning the HDEV may be mapped as a LUN within multiple different Host Storage Domains on the same or different ports.

- **MP** (microprocessor) – the NEC Y-wing 800MHz CPU on USP V FED and BED boards.
- **MPU** (logical processor) – the name of the set of four cores on the Intel Xeon quad-core 2.33GHz CPU used on the VSD processor board. LDEV ownership is per MPU.
- **Parity Group** (configurable, **RAID Group**) – a set of one or two storage system Groups (a set of 4 or 8 drives) formatted as a single RAID level, either as RAID-10 (often referred to as RAID-1+ in HDS documentation), RAID-5, or RAID 6. Internal LDEVs are carved from the VDEV(s) corresponding to the formatted space in a Parity Group, and thus the maximum size of an internal LDEV is determined by the size of the VDEV that it is carved from. The maximum size of an *internal* VDEV is approximately 2.99 binary TB. If the formatted space in a Parity Group is bigger than 2.99 TB, then multiple VDEVs must be created on that Parity Group. Note that there actually is no discrete 4+4 Parity Group type - see *Concatenated Parity Group*.
- **PCIe (PCI Express)** – a serial bus connection technology that supports x1, x4, and x8 link configurations. The VSP uses the 1.0 x4 version in the HiStar-E Grid. The x4 link is capable of 1GB/s send plus 1GB/s receive in full duplex mode (i.e. concurrently driven in each direction). On PCIe 1.x, each “lane” is capable of up to 2.5 Gbit/s simultaneously in each direction. This is 250 MB/s of data and commands in each direction (full duplex mode). Note that 10 bits are used to encode an 8-bit byte as it is transferred over the link. See Appendix 10 for more details.
- **PDEV (Physical DEVICE)** – a physical internal drive.
- **RAID-1 or RAID-1+** – used in some documents to describe what is usually called “RAID-10”, a *stripe of mirrored pairs*. Thus when we say “RAID-1” in the context of a Hitachi enterprise VSP-family storage system, we mean the same thing as when we say “RAID-10” in the context of an AMS modular storage system. Note that RAID-0+1 is quite different, as it is the very vulnerable *mirror of two stripes*.
- **SBX** – the small form factor DKU (128 slot drive container)
- **Shared Memory (USP V only)** - the two or four special purpose memory boards in the USP V design that contain the Control Memory regions, accessible by each FED or BED using dedicated 250MB/s serial links to these boards. All four boards needed to be installed in a system in order to enable all eight of the Shared Memory paths present on each FED or BED board.
- **SVP** (Service Processor) - a blade PC running Windows Vista Business that is installed in the control frame and used for system configuration and reporting.
- **UBX** – the large form factor DKU (80 slot drive container)
- **VDEV** (configurable) – The logical storage container from which LDEVs are carved. There are four types of VDEVs:
 - o **Internal** VDEV (2.99 binary TB max): maps to the formatted space within a parity group that is available to store user data. LDEVs carved from a parity group VDEV are called internal LDEVs.
 - o **External** storage VDEV (4 binary TB max): maps to a LUN on an external (virtualized) storage system. LDEVs carved from external VDEVs are called external LDEVs.

- o **Copy on Write (CoW)** VDEV (4 binary TB max): called a "V-VOL group", and LDEVs carved from a CoW V-VOL group are called a CoW V-VOLs
- o **Dynamic Provisioning (DP)** VDEV (60TB max): called a "V-VOL group", and LDEVs carved from a DP V-VOL group are called a DPVOLs (Dynamic Provisioning Volumes).
- **VSD (installable, Virtual Storage Director)** - the processor board (2-8 per storage system) that contains an Intel quad-core Xeon (**MPU**) and local DRAM for its core workspace and a partial (working) copy of Control Memory for just those LDEVs managed by that MPU. All I/O operations are managed by these boards, and they control the operations of the FED and BED boards. Each VSD (MPU) has a limit of 256k tags that it can manage.
- **V-VOL Group** (configurable) – either a Dynamic Provisioning VDEV or a Copy on Write VDEV. With Dynamic Provisioning, it is used to hold one DPVOL. V-VOL Groups are used to parallel the internal architectural structures of internal storage.

Basic Theory of Operation

In the VSP (like all Hitachi designs), there are separate intelligent components from which the storage system is created. These components are operated in parallel to achieve high performance, scalability, and reliability.

Briefly (the boards are discussed later in greater detail), a VSP is configured as a either single control chassis or dual control chassis system, where each control chassis is associated with one to three frames. Note that the following usage is commonly used to describe the storage system:

- **Single chassis** – meaning a control frame (DKC) and 0, 1, or 2 expansion disk frames supporting 0 to 8 DKUs (it may also be diskless – no DKUs) or FBXs (1 or 2)
- **Dual chassis** – meaning two control frames, and 0-4 expansion disk frames supporting 1 to 16 DKUs (there must be at least one DKU) or FBXs (1-4)

A **single chassis** system will have one control frame and up to two drive expansion frames. The control frame (DKC) has the logic box (in the bottom section) that holds all of the control boards (shown in Figure 1) for that storage system, along with zero, one or two drive container units (DKUs or FBXs). The disk expansion frames can hold three DKUs each. There are two types of DKUs: Small Form Factor (128 2.5" drives) and Large Form Factor (80 3.5" drives), as well as one type of FBX (48 FMDs).

A **dual chassis** system will have two control frames, one to four drive expansion frames, and one to 16 DKUs (including up to four FBXs). The two control chassis are externally cross connected at the Grid Switch level (with 8 or 16 paths) to form a single integrated storage system. The storage system behaves as a shared system image, not as a pair of units operating as a cluster. Note that the BEDs in each chassis only interact with those local drives in that chassis's DKUs or FBXs.

The VSP uses five types of adapter boards in the logic box:

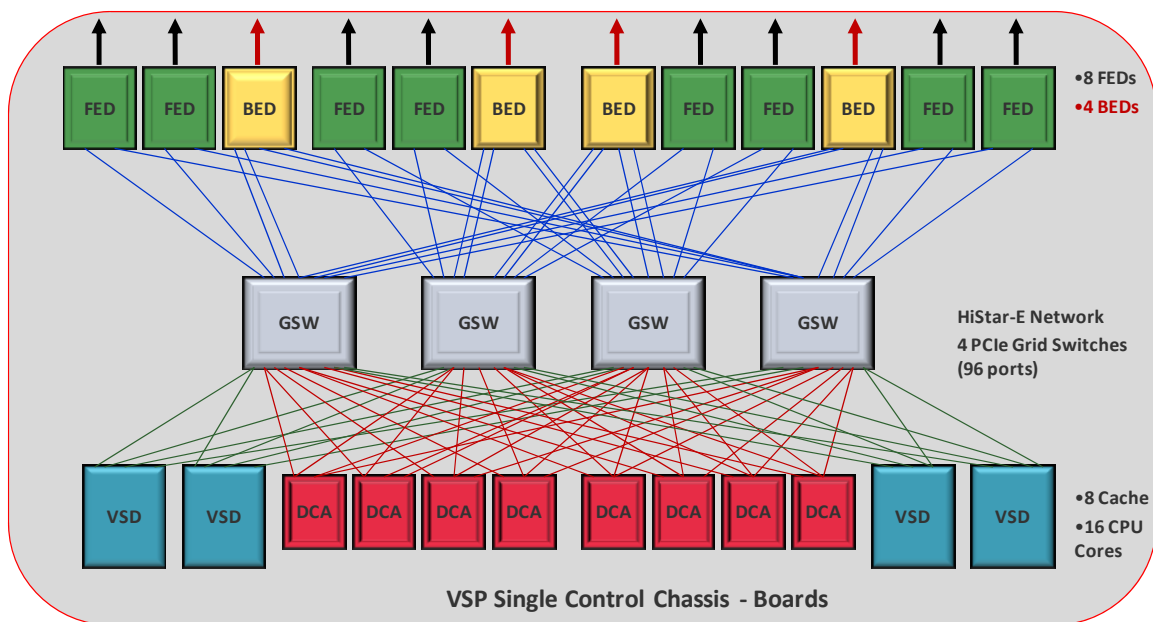
- Grid Switches (**GSW**)
- Data Cache Adapters (**DCA** cache boards)
- Front End Directors (**FED** FC, FCoE, or FICON ports)
- Back End Directors (**BED** drive controllers)

- Virtual Storage Directors (**VSD** processor boards, or **MPU** – the logical processor).

The following sections provide a brief discussion of how these boards work together in the VSP design. Later in this paper, there are sections providing details on each of these boards. Appendix 7 also shows a fully configured USP V storage system shown in the same illustration style for comparison.

For every host I/O request, all FED or BED metadata and job control traffic goes to the MPUs on the Virtual Storage Director boards. However, the FED or BED boards move user data blocks directly in or out of cache (DCA boards) with DMA transfers to locations specified by the MPUs. The set of VSD boards direct all of the work done in a storage system (whether a single or dual chassis configuration) and manage all cache usage.

Figure 1. A VSP Single Chassis Storage system Fully Populated with All 28 Boards



While the VSD board MPUs each manage a set of dynamically allocated 64KB cache segments (with a global free set of 64KB segment available to all of the MPUs) for user data, the MPUs themselves do not have any access to these same segments. Similarly, only the MPUs have access to the reserved Control Memory region in cache memory (on the first pair of DCA boards in a storage system). There are hardware address interlocks that prevent VSD boards from accessing the data region of cache, and the FED and BED boards are similarly prevented from accessing Control Memory and the Cache Directories.

Architecture Overview

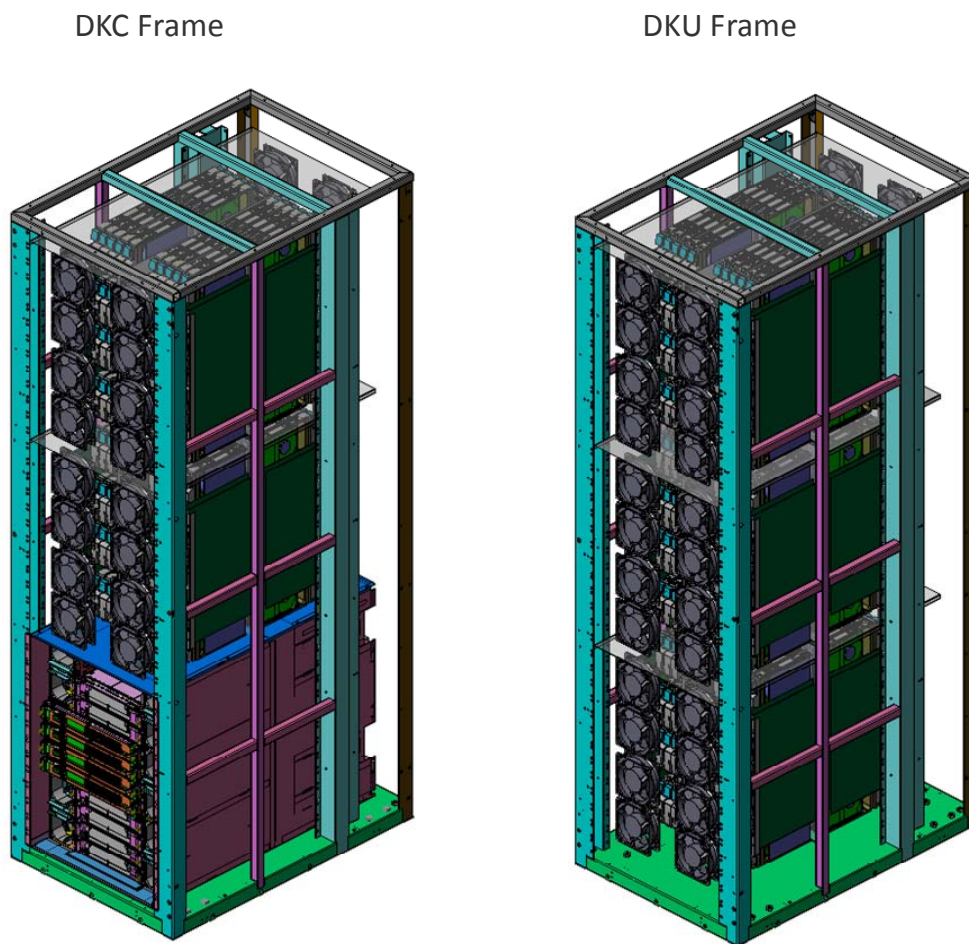
The VSP storage system can be configured as a single chassis (CBXA) or dual chassis (CBXA + CBXB). Each chassis has at least one frame (rack), the control frame (or DKC) in which there is a logic box for controller boards plus one or two optional drive containers (DKUs or FBXs). A single chassis can be expanded by one or two drive expansion frames, each with up to three DKUs. A DKU is either of the Small Form Factor (2.5" drives) or Large Form Factor (3.5" drives) type. An FBX is for up to 48 of the FMD drives (two per CBA and two per CBXB).

A dual chassis storage system will have two control frames and up to four drive expansion frames (or six frames in all). The logic boxes in each chassis are cross connected at the Grid Switch level (using either 8 or 16 PCIe 1.0 x4 paths) to create a single system image (not a cluster pair). A fully configured dual chassis (six-rack) storage system occupies a small footprint (3.6 feet by 11.8 feet) and consumes much less power than a fully configured USP V design.

The VSP has no specific models as such, but starts with a single basic control frame (CBXA) onto which upgrade features are applied to meet the needs of the customer. A second chassis (CBXB, fully integrated into a single uniform storage system) may be added in order to scale out the overall connectivity, storage capacity and performance.

Note: One cannot merge two operational single chassis storage systems into one dual chassis system, or break apart a dual chassis storage system into two single chassis storage systems and retain the configuration or data. All metadata and configuration information only exists in the first chassis in its Control Memory region. All hardware and configuration data must be redone via the primary chassis as if it is all new hardware. Also, if the second chassis is accidentally powered down it will immediately cause an emergency shutdown of the first chassis (a fatal environmental error).

Figure 2. Control Frame and drive Expansion Frames



Frames

There are no common parts between the VSP and the previous USP V model. The manner in which they operate internally is very different. As such, there are significant differences with storage system planning and deployment at a customer site, as briefly discussed later in the Summary section.

As already described, a VSP storage system is either configured as a single or dual chassis collection of control frames, drive expansion frames, FBXs or DKUs (disk containers with 8 HDUs), logic boards for the logic box, cache memory, and drives.

A **single chassis VSP** can include up to the following:

- 1 control frame (with a logic box) and 2 drive expansion frames
- 4 GSW grid switch boards
- 4 VSD Xeon quad-core processor (MPU) boards
- 8 FED boards with 8Gbps FC (64 ports), 10Gbps FCoE (32 ports), or FICON (64 ports)*
- 4 BED boards with 32x 6Gbps back end SAS links
- 8 DCA boards with 512GB of cache (using the 8GB DIMMs)
- 8 SFF and LFF DKU boxes (with 8 HDUs each), or 6 DKUs and 2 FBXs
- 1024 2.5" drives (or 640 3.5" drives) – in 64 HDUs, including 96 FMD devices in 8 FMUs

A **dual chassis VSP** can include up to the following limits:

- 2 control frames (2 logic boxes) and four drive expansion frames
- 8 GSW grid switch boards
- 8 VSD Xeon quad-core processor (MPU) boards
- 8 BED boards with 64 6Gbps back end SAS links
- 16 FED boards with 8Gbps FC (128 ports), 10Gbps FCoE (64 ports), or FICON (128 ports)*
- 16 DCA boards with 1024GB of cache (using the 8GB DIMMs)
- 16 SFF and LFF DKU boxes (or 12 DKUs and 4 FBXs)
- 1280 to 2048 drives (depends on DKU form factor) – in 128 HDUs, including 192 FMDs in 16 FMUs

*More FED boards may be installed if some BEDs are not used.

For Figures 3 and 4, the legend below shows the Chassis (DKC-0 or DKC-1) and the BED feature (1 to 4) ownership through each DKU (128 or 80 drives each). Also, containers DKU-00, DKU-01, DKU-10, and DKU-11 could instead be FBX-00, FBX-01, FBX-10, and FBX-11 (Accelerated Flash containers).

Figure 3. Frame Configuration Map of DKUs and HDUs (Front view)

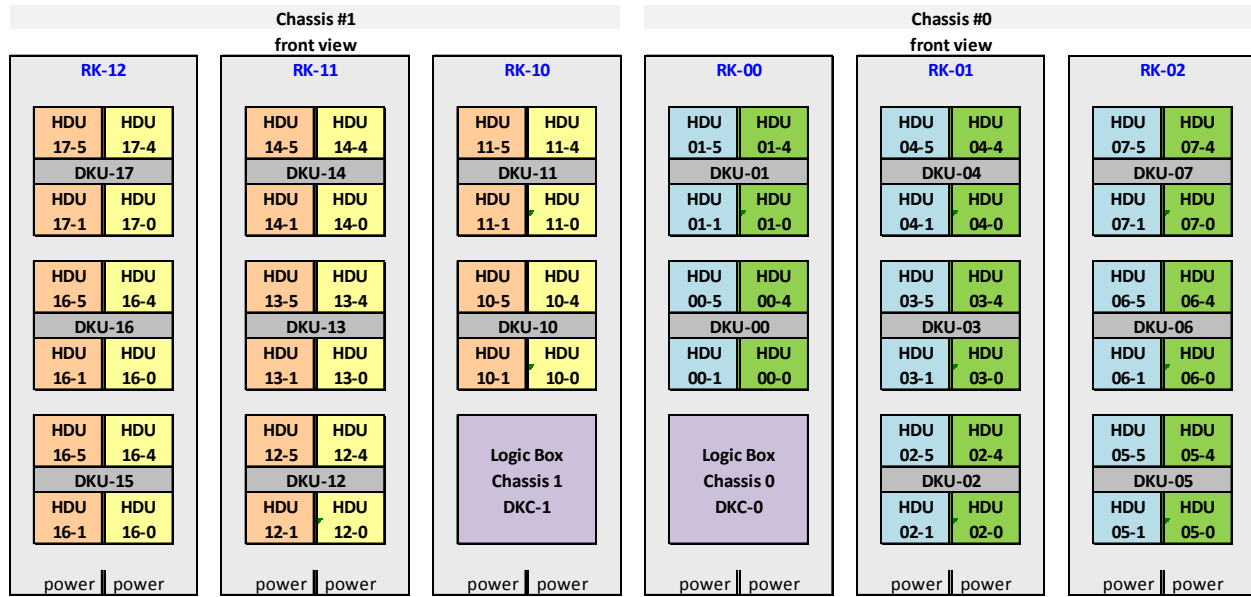


Figure 4. Frame Configuration Map of DKUs and HDUs (Rear view)

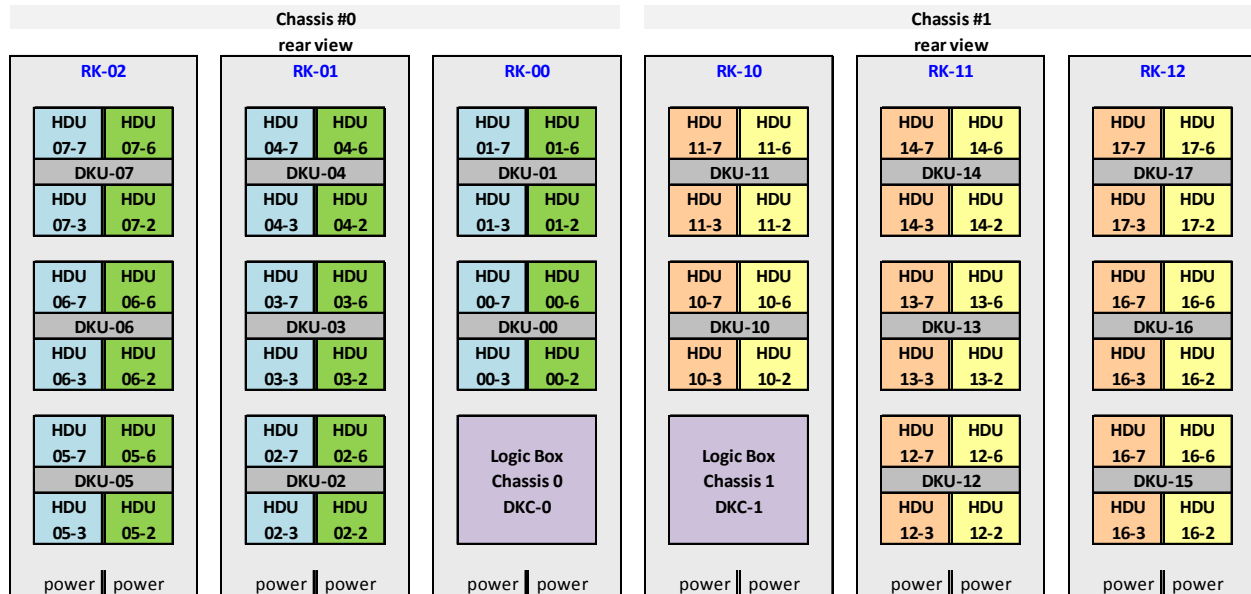


Figure 5. Legend of DKC and BED Ownership of DKUs

DKC-0		DKU #							
BED 1, 2	0	1	2	3	4	5	6	7	
	0	1	2	3	4	5	6	7	

DKC-1		DKU #							
BED 3,4	10	11	12	13	14	15	16	17	
	10	11	12	13	14	15	16	17	

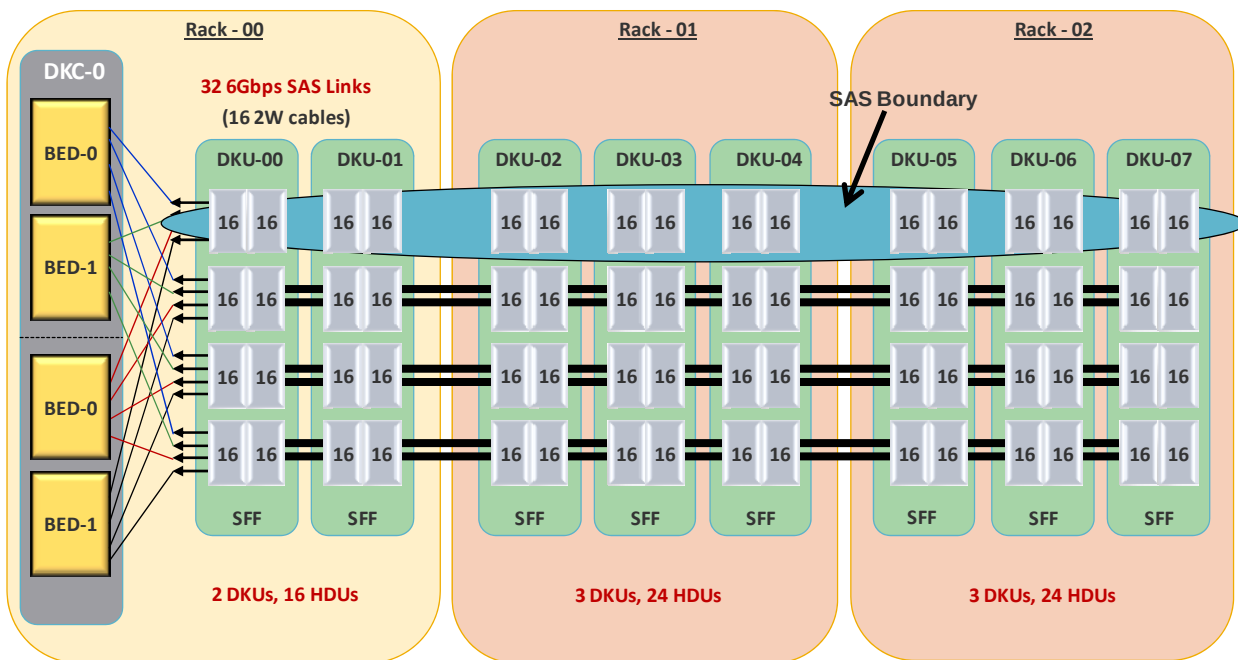
DKUs and HDUs

Each of the frame mounted DKU boxes per chassis (1 to 8) has eight internal **HDU** drive containers. If the DKU box is the **SFF** type, it can hold **128 drives**, with 16 drives per HDU. If it is the **LFF** type, a DKU can hold **80 drives**, with 10 drives per HDU. Drives are added to four specific HDU slots within the same DKU in sets of four. Array Groups are installed (following a certain upgrade order) into specific HDU drive slots in a DKU logical region known as a “**B4-x**” (see Appendix 6 for these maps). Each “**B4-x**” is the set of four HDUs either from the top horizontal section or the bottom horizontal section of each DKU box (as shown later in the *SAS Engine Concepts* section).

Note that all of the HDUs within a single DKC chassis are controlled by the two or four BED boards (as one or two option features) installed in that same chassis. A second DKC chassis, configured as part of that same storage system, would have its own separate BEDs and DKUs.

The overall BED and DKU layout for a single DKC system is illustrated in Figure 6. The green boxes are the DKUs (all 8 shown) and the gray boxes within the green boxes are their 8 HDUs. A **standard performance** DKC uses only one BED feature (BED-0, with 2 boards and 8 SAS ports), which provides 16 x 6Gbps SAS links from the BEDs to the one-to-eight DKUs. The **high performance** DKC uses two BED features (adds BED-1, with 4 boards and 16 SAS ports) to provide a total of 32 x 6Gbps SAS links. In each case, with all eight optional DKUs installed, there may be up to 1024 SFF drives or 640 LFF drives in a single control chassis system.

Figure 6. Overview of the Dual BED to SFF DKU Organization in a Single DKC Chassis (HDUs are in gray)



- Up to 1024 SFF (shown) or 640 LFF disks, 32 600MB/s SAS links (16 2W ports), 8 DKUs, 64 HDUs

Note that each of the four lines shown per BED board (as shown in Figure 6) is actually a SAS 2W (2-Wide) cable containing two independent, full duplex 6Gbps SAS links. Thus, on a single chassis with all four BED boards installed, there will be 8 x 6Gbps SAS links passing through each horizontal “stack” of HDUs (across the installed DKUs). This is a **SAS Boundary**. Thus there are four SAS boundaries per single chassis system, and 8 SAS boundaries per dual chassis system. These 8 SAS links control up to 256 drives

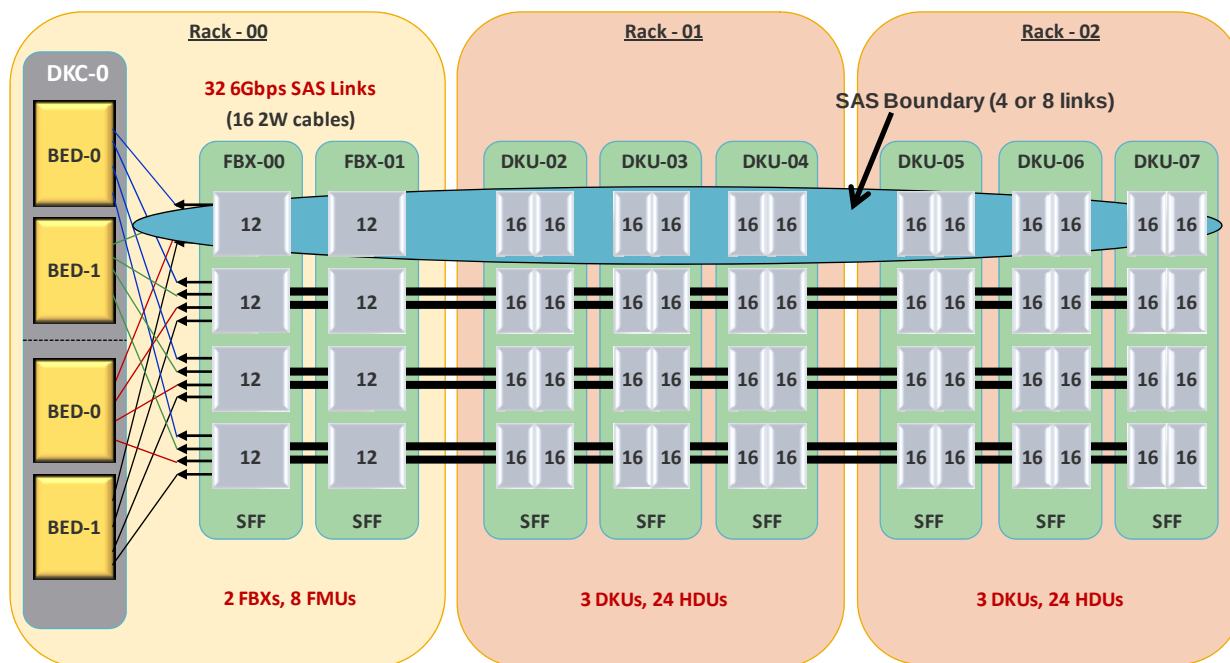
(SFF) or 160 drives (LFF) from 16 of the 64 HDUs (with eight DKUs in a maximum configuration) within a chassis.

Each FMD drive is quad-ported, each SAS and SSD drive is natively dual ported, and each SATA drive is single ported. The individual SATA HDU canisters have an adapter board that connects the single-ported SATA drive to the each of the two SAS expander boards within a DKU. As SAS or SSD drives have two full duplex ports, they support four communications with the SAS controller at a time (1 data transfer operation and up to three command operations). [An FMD drive is expected to support double the SAS communications rate.] The SATA drives are single ported and half duplex, so they only support one communication at a time even with the “Y” adapter board in the drive canister.

FBXs and FMUs

An FBX is the drive container for the proprietary Hitachi Accelerated Flash module (drive) feature. Each of the frame mounted **FBX** boxes per chassis (1 or 2) has four internal **FMU** drive containers. Each FMU container holds up to 12 **FMD Accelerated Flash** modules. Each FMU is connected to four of the BED SAS ports as shown in Figure 7 below. With two FBXs installed, each with 48 FMD drives, each SAS Boundary would include 24 drives (plus any other drives in HDUs from DKUs). The FBX boxes should be installed closest to the BEDs (as shown below) to achieve highest performance (avoid the latency of passing through the many DKU switches in front of them). A second control chassis (CBXB) configured as part of that same storage system would have its own independent BEDs and one or two additional FBXs.

Figure 7. Overview of the Dual BED to FBX and SFF DKU Organization in a Single Control Chassis System



Logic Box and Logic Boards

There are five types of **logic boards** available for use in the VSP logic boxes. One of these is the high performance Grid Switch board that interconnects the other five types of boards together. This central Grid Switch system is known as the **HiStar-E Network**. Figure 8 provides an overview of the individual logic boards in a fully populated logic box for a single DKC chassis. A fully populated second DKC chassis would duplicate this figure. Not shown are the four links per GSW that connect it to the corresponding GSW in the other chassis. There are no cross-connections among the GSWs within a chassis.

Option boards are installed in the front and rear slots in the **Logic Box**. Figures 9 and 10 illustrate the locations and names of these slots in DKC-0 and DKC-1. The logic board choices for the VSP include the following options (known as **features**, each a pair of boards installed on the two power domains). Both the Hitachi Data Systems name and the Hitachi factory name are included:

- **GSW** Grid Switch (also **ESW**), a PCI Express Switch – 1 or 2 features (2 or 4 boards) per chassis each with 24 2GB/sec (the full duplex rate) HiStar-E ports.
- **DCA** Data Cache Adapter (also **CMA**), the cache memory board – 1 to 4 features (2, 4, 6 or 8 boards) per chassis with up to 64GB of cache each (using 8GB DDR3 DIMMs), with 8GB (2 x 4GB DIMMs) being the minimum.
- **VSD** Virtual Storage Director (also **MPB**), the processor board - 1 or 2 features (2 or 4 boards) per chassis each with 1 quad-core Intel Xeon CPU and 4GB of local DDR2 DRAM.
- **FED** Front end Director (also **CHA**), the host port board – 1 to 4 features (2, 4, 6, or 8 boards) per chassis: **8 x 8Gbps FC** boards, **4 x 10Gbps FCoE** boards or **8 x 8Gbps FICON** boards.
- **BED** Back end Director (also **DKA**), the drive controller board – 1 or 2 features (2 or 4 boards) per chassis with 8 x 6Gbps SAS links per board. If the BED features are not installed (this is only available for a single chassis configuration), then two additional FED features may be used in those BED slots.

Figure 8. Overview of Boards in a Fully Populated Single Chassis Logic Box

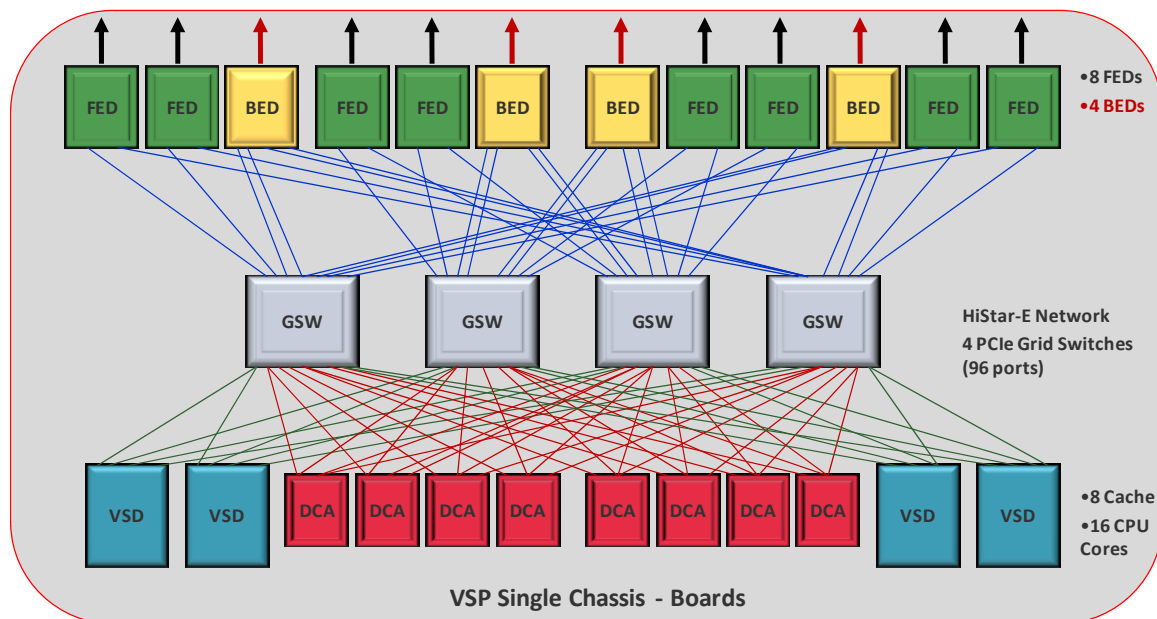


Figure 9. Logic Box Slot Names (for Chassis #1)

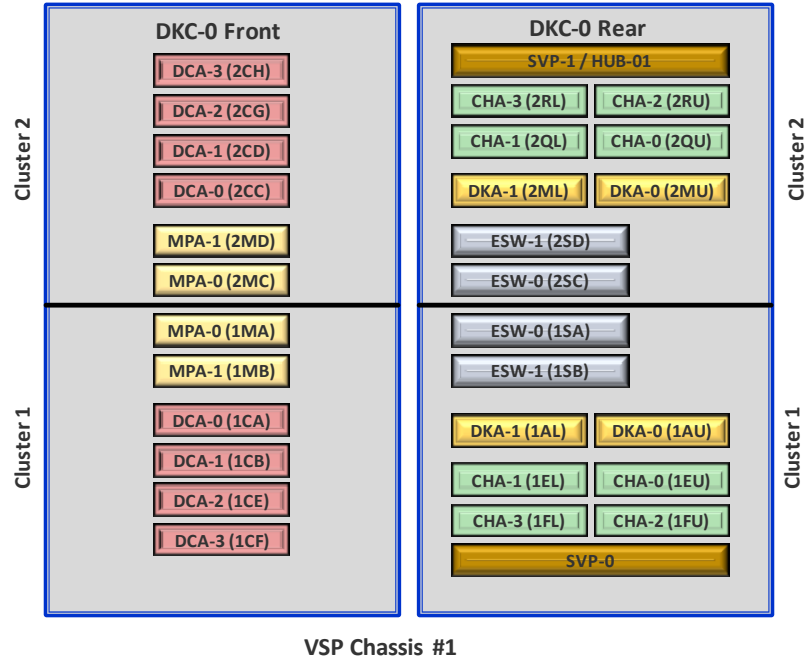
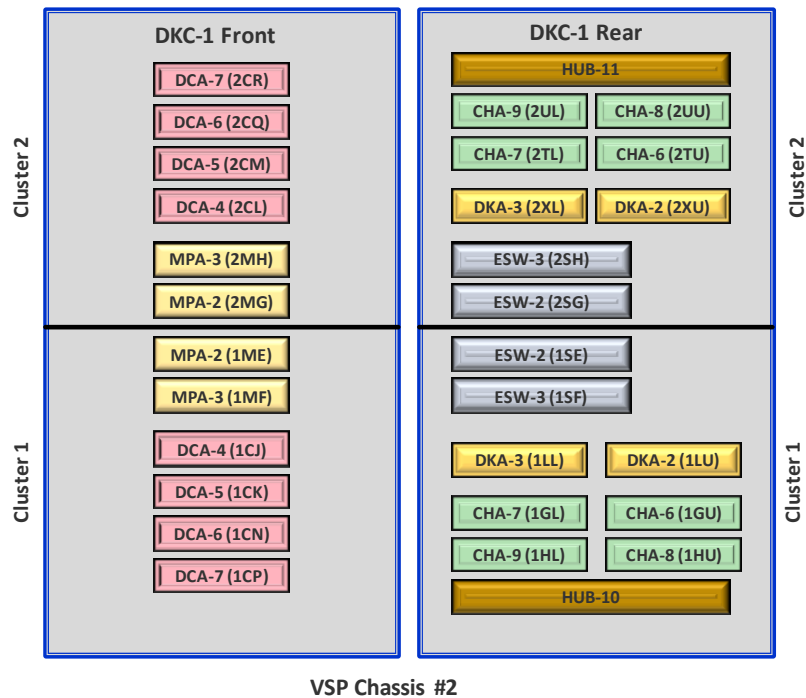


Figure 10. Logic Box Slot Names (for Chassis #2)



Array Limits

Table 1 below shows the major limits for the VSP (both single and dual chassis) as well as a maximally configured USP V shown for comparison.

Table 1. Summary of Limits, Virtual Storage Platform to Universal Storage Platform V

Table of Limits	VSP		USP V
	Single Chassis	Dual Chassis	Maximum
Data Cache (GB)	-	-	512
Shared Memory (GB)	-	-	12+12
Physical Cache (GB) - VSP	512	1024	-
Control Memory (GB), including mirror	28+28	28+28	-
Cache Directories	2 to 4GB	4 to 8GB	-
Raw Cache Bandwidth	64GB/s	128GB/s	68 GB/sec
FMD Drives	96	192	-
SSD Drives	128	256	128
2.5" SAS drives	1024	2048	-
3.5" SAS drives	640	1280	-
3.5" FC drives	-	-	1152
SATA drives 3.5"	640	1280	1152
Logical Volumes	65,280		65,280
Logical Volumes per VSD board	16,320		-
Max Internal Volume Size	2.99TB		2.99TB
Max CoW Volume Size	4TB		4TB
Max External Volume Size	4TB		4TB
Maximum tags per VSD board	256k		-
IO Request Limit per Port (rule of thumb)	2048		2048 fixed
Queue Depth per LUN (rule of thumb)	32		32
HDP Pools	128		128
Max Pool Capacity	4PB		1.1PB
Max Capacity of All Pools	5PB		1.1PB
LDEVs per Pool (pool volumes)	1024		1024
Max Internal Pool Volume size	2.99TB		2.99TB
DP Volumes per Pool	~62k		8192
DP Volume Size Range (No SI/TC/UR)	46MB-60TB		46MB-4TB
DP Volume Size Range (with SI/TC/UR)	46MB - 4TB		46MB - 4TB

Logic Board Details

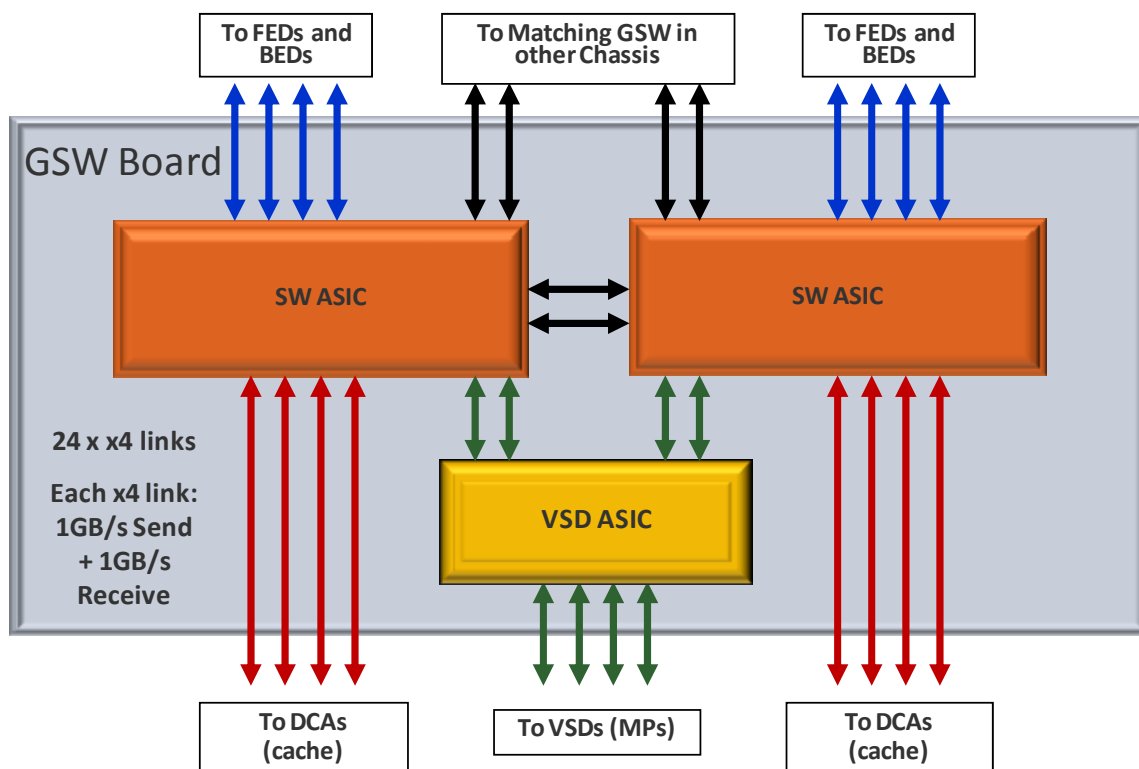
This section describes each of the five types of controller boards in detail.

HiStar-E Grid Switch (GSW)

Overview

At the core of the new VSP architecture is the **HiStar-E Network**, a PCI Express **Grid Switch** system. The HiStar-E Network is a fully fault tolerant, high performance, true non-blocking switched architecture. All of the other boards are attached to these switches within the logic box. When a dual chassis storage system is configured, there are cross connections (shown here as the external black links) between the corresponding Grid Switches in each logic box.

Figure 11. Grid Switch Board Details



There are either two or four GSW boards per control chassis. In order to install more than two FED features and one BED feature all four GSWs are required. Each GSW board has **24 full duplex 2GB/s links (ports)**, with each PCIe x4 link able to concurrently send at 1GB/s and receive at 1GB/s. As such, the GSW supports an aggregate peak load of 24GB/s send and 24GB/s receive. These 24 links are used as follows:

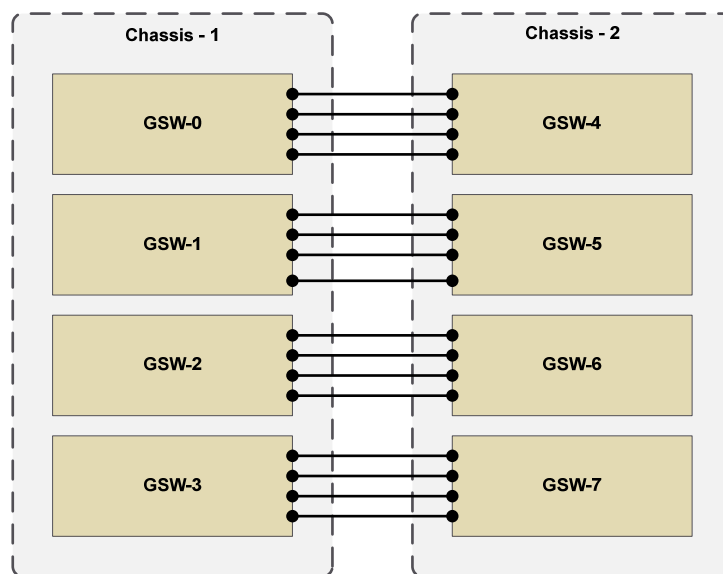
- Eight links (**blue**) are dedicated for connection to FED and BED ports. These ports see both data and metadata traffic intermixed.
- Four links (**green**) are dedicated for connection to VSD boards. These ports only see job request traffic or system metadata (such as the address in cache for a target slot).
- Eight links (**red**) are dedicated for connection to DCA boards, which see Control Memory updates as well as user data reads and writes.

- Four links (**black**) are dedicated for the cross connections from one GSW board to the matching GSW board in the second chassis (if used).

There are no cross connections among the GSWs within a chassis. Every FED and BED board in a chassis attaches to two GSWs, while each VSD and DCA board attaches to all four GSWs.

Figure 12 shows how the individual GSW boards are cross attached with x4 links (four per matched pair) in a dual DKC chassis configuration. This is a one-to-one association, not one-to-many. One could view these connections as being somewhat like ISL ports on a SAN switch.

Figure 12. GSW Cross Connections between the Control Chassis



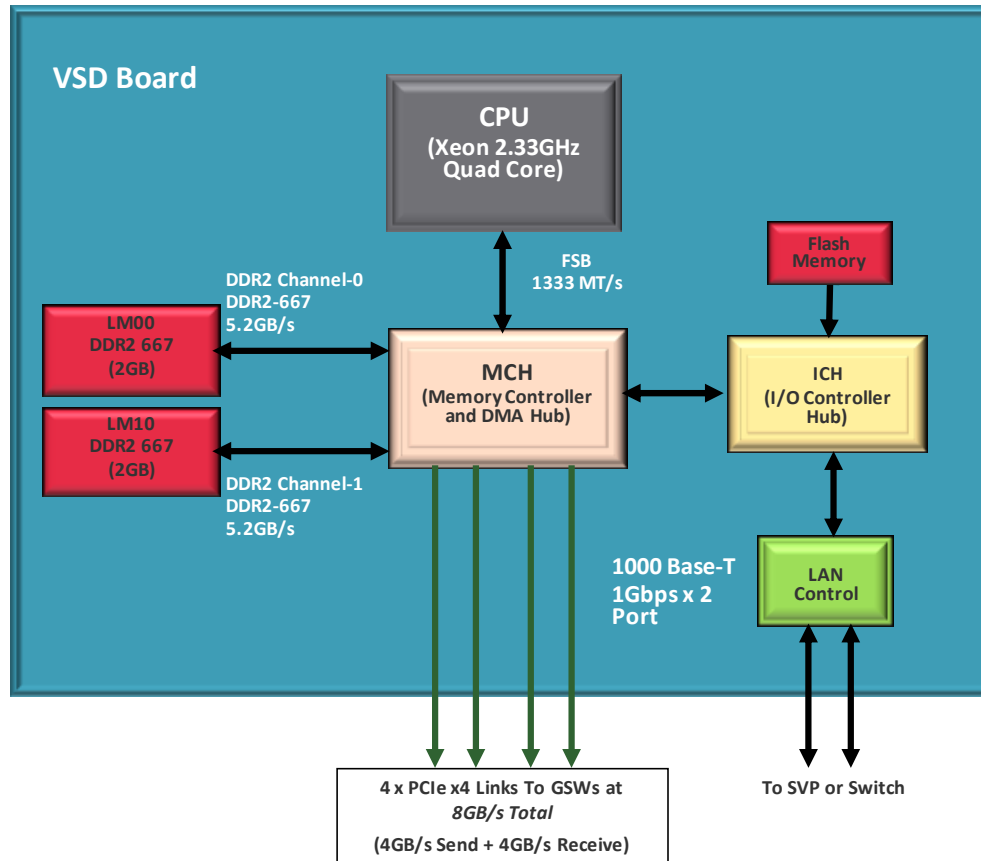
Virtual Storage Director Board (VSD)

Overview

The **Virtual Storage Director (VSD)** is the I/O processor board. There are either two or four VSD boards installed per single chassis storage system (so up to 8 boards per dual chassis storage system). The addition of the optional pair of VSD boards increases the random I/O power of the storage system. This is separate from the amount of host ports or drive paths. The VSD processors (**MPU**) tell the FEDs and BEDs what they have to do and where in cache to operate.

Each VSD board contains a quad-core Xeon CPU (referred to as the **MPU** in many documents) and 4GB of local DRAM. These boards execute all of the internal I/O jobs dispatched from the DA processors on the FEDs and BEDs. It is important to note that user data is never passed through the grid links to a VSD board. The MPUs only execute host I/O requests (identified by up to 256k “tags” per board) and software such as HDP or copy products, as well as manage the Control Memory tables. There is a Flash Memory device on each VSD board which contains the system microcode and current configuration. These Flash Memory devices are the “boot devices” for the storage system.

Figure 13. Processor Board Details



The firmware loaded into the VSD board's Flash Memory device contains all of the types of previously separated code on the USP V. Each of the four MPU cores will schedule a process that depends upon the nature of the job it is executing. A process will be of one of the following types or their mainframe equivalents: **Target**, **External (virtualization)**, **BED**, **MCU**, or **RCU**. In addition there will be storage system software such as **UVM** or **HDP** and housekeeping processes executed by the various MPUs.

- **Target** process – manages host requests to and from a FED board for a particular LDEV.
- **External (virtualization)** process – manages requests to or from an FC FED port used in virtualization mode (external storage). Here the FC FED port is operated as if it were a host port in order to interoperate with the external storage system.
- **BED** process – manages the staging or destaging of data between cache blocks and internal drives via a BED board. The DRR functions within the FED and BED DA chips (discussed later) process parity for writes (not handled by the VSD MPUs).
- **MCU initiator** process – manages the “respond” side of a Hitachi Universal Replicator connection on a FED port, or the “send” side of a TC connection.
- **RCU target** process – manages the “pull” side of a remote copy connection on a FED port, or the “receive” side of a TC connection.

Each MPU communicates over four PCIe x4 links into the switched grid network to the Data Accelerator processors located on each FED or BED board. They do not directly communicate with hosts, drives or the data in cache. One very simplistic manner of visualizing this separation of responsibilities is that the VSD board has an “MPU server” running threads, the FED is a host port card, and the BED is a drive controller card, and the FEDs and BEDs perform DMA transfers to memory. The Data Accelerator processor (on FEDs and BEDs) functions as a processor and a data pump to move data between the host and cache (FEDs) and between the cache and the drives (BEDs) under the direction of the VSD MPUs.

Each VSD board has local DRAM that is partitioned into five spaces:

1. private work spaces for MPU core-1
2. private work spaces for MPU core-2
3. private work spaces for MPU core-3
4. private work spaces for MPU core-4
5. a shared Control Memory region

All metadata associated with those LDEVs managed by that MPU is located in this local Control Memory space. All lookups occur in here at very high speed. All updates to the metadata for these LDEVs *write through* local memory over a GSW link into the master copy of Control Memory located in cache (the first two DCA boards). This new design is much quicker (lower latency) than the previous USP V Shared Memory system. Since only one VSD controls all I/O to an LDEV, there are no cross-MP coherency locks as required in the USP V’s Shared Memory model. MPUs do not work together but operate totally independently.

As user data is not processed within the VSD board itself, no GSW bandwidth is needed for user data across its four GSW links. A more detailed discussion of the Control Memory mechanism and user data cache is found in the **Memory Systems** section below.

LDEV Ownership by VSD (MPU)

Each VSD board MPU (4-core processing element) is assigned certain LDEVs to manage. There is a 65,280 LDEV storage system limit whether single or dual chassis. [*Note - there are many types of VDEVs that LDEVs point to – see Appendix 4.*] Only one of the installed VSD boards manages the I/O for an individual LDEV. This strict LDEV ownership by MPU reduces the competition for access to Control Memory and data blocks in cache to updates.

When Parity Group volumes (LDEVs) are created, they are round-robin assigned across all of the MPUs in a system whether it is a single or dual chassis configuration (it is still one unified system). In a single chassis storage system with four VSD boards (two features) each MPU will start out managing roughly 25% of all defined LDEVs in the storage system. In a dual chassis configuration with 8 VSDs, each VSD will start out managing 12.5% of all LDEVs.

The individual LDEV associations to MPUs can be looked up and manually changed in Storage Navigator or by a script that uses CLI commands from the **raidcom** utility. There is no automatic load balancing mechanism to move “hot” LDEVs around among the VSDs to even out the MPU loads. The exception is that, when adding a new VSD feature, the system will automatically reassign some LDEVs to the new boards in order to balance out the distribution (unless a system mode is set that blocks this).

An MPU will accept all I/O request for an LDEV it owns without regard for which FED board handled the host request, or for which BED is the target of the physical I/O. Each MPU interacts with all installed FEDs and BEDs (across both chassis). Adding VSD boards to a storage system simply increases the MPU

processing power (I/O jobs and software execution) of that storage system. In general, having more VSD boards primarily benefits small block random workloads. The addition of a VSD feature per chassis will not increase sequential throughput (but adding cache boards and BED boards will do so). The number of installed VSDs is unrelated to the number of installed FEDs or BEDs.

Each FED board maintains a local copy of the LDEV-to-MPU mapping tables in order to know which MPU owns which LDEV. No other MPUs will be involved in these operations unless the owning VSD board fails. If a VSD board should fail, all of its assigned LDEVs are temporarily reassigned to the other VSD board in that same feature (installed pair). The 16K LDEV limit of the VSD board will “relax” to 32K, but this will require paging in/out of that VSD’s local memory to Control Memory if the total number of mapped LDEVs is now greater than 16K. The fail-over VSD will read all of the necessary Control Memory information from the shared master Control Memory tables maintained in the cache system. Upon replacement of the failed VSD board, the reverse will occur, and the original LDEV management will revert to the replacement board.

As a rule of thumb, an MPU should be kept below 70% busy to manage host latencies, and only 35% busy if providing for processor headroom to maintain host performance in case of a failure of one VSD board of a pair. Note: higher peak utilization during batch operations is fine as long as the average utilization during the time remaining in the batch window is below 40%.

VSD Command Tags

The MPU command tag buffers are located on each VSD board. Each LDEV has affinity to one MPU. Microcode and metadata processing is executed on the cores of each MPU on behalf of work associated to the LDEVs assigned to each VSD board. Each board has buffer space containing 256K command tag entries. Each VSP can scale the VSD boards to eight (8) meaning a VSP has a total system wide command tag capacity of 2,048K host command tag entries.

Since the command tags are located on the global VSD boards, the FC and FCoE ports have no real limits in queue depth (tag limit) other than the system wide limit of 2,048K command tags, if the active LUNs on any given port are distributed across all 8 VSD boards. Also, a LUN’s maximum queue depth (tag limit), regardless how many ports have access to it, is equal to maximum command tags of a single MPU (256K command tags per VSD board).

The command tags entries of each MPU are dynamically assigned and reassigned to active LDEVs. If a VSD board has only one active LDEV, then that LDEV theoretically has a tag limit of 256K. If a second LDEV becomes active on that VSD board, then each active LDEV have a tag limit of about 128K, and so on.

So rather than assigning the tag limits in a distributed fashion at the port level as for past generations of HDS arrays and current competitors arrays, the HDS VSP assigns the command tags globally to the system, thus reducing the potential of exhausting port level command tags while other ports have unused command tags. This is the same enterprise approach to other global system wide resources in the VSP.

Data Cache Adapters

Overview

The **Data Cache Adapter (DCA)** boards are the cache boards that buffer all user data and contain the master copy of Control Memory (metadata). There are up to 8 DCA boards (four features) installed per

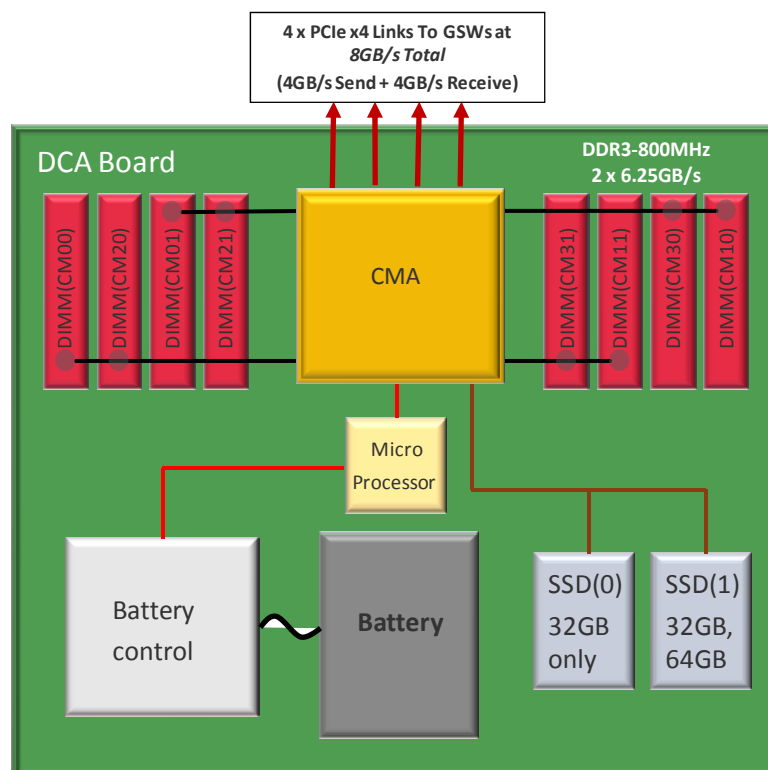
single chassis, providing up to 512GB of cache when using the 8GB DIMMs. A dual chassis storage system can have up to 1024GB (8GB DIMMs) of cache on 16 DCA boards.

Each DCA cache board has eight **DDR3-800 DIMM** slots, organized using four memory controllers with 2 DIMM slots each (thus 32 independent banks of DIMMs in a single chassis storage system with 8 DCA boards). The pair of boards for a DCA feature is installed in different power domains in the Logic Box (Cluster-1 and Cluster-2). Each DCA board can support 8-64GB of DRAM using the 4GB or 8GB DDR3 DIMMs. The same amount of DRAM must be installed on each DCA board of a feature pair, but may differ among the installed DCA features. However, not having equal amounts of RAM installed on all DCA boards means uneven loading on the DCA boards and GSW links due to the capacity mismatch.

Each DCA board has four PCIe x4 full duplex GSW links, each operating at a **1024MB/s send** and **1024MB/s receive** rate (concurrently). This provides for a theoretical 8GB/s of aggregate read-write bandwidth per board to the FEDs, BEDs, and MPUs (Control Memory, Cache Directories). Each bank of DDR3-800 DRAM has a peak bandwidth of 6.25GB/s, or a maximum of 25GB/s possible per board. Due to the very high speed of the DRAM compared to the GSW link rates, all four GSW links can operate at full speed.

The first two DCA boards in the base control chassis (but not in the second control frame) have a region of up to **56GB** (28GB per DCA board, mirrored) used for the storage system's master copy of **Control Memory**. Every DCA board also has a **500MB** region reserved for a **Cache Directory**. This is a mapping table to manage pointers from LDEVs and allocated cache segments to those LDEVs using that cache board. The remainder of the space is evenly allocated to the MPUs as 64KB segments for data buffers.

Figure 14. Data Cache Adapter Board



SSD Drives for Backup

Each DCA board has one or two local SSD drives installed (32GB or 64GB decimal, or 31.5GB and 63GB binary as used by RAM) for use in backing up the entire cache space in the event of a storage system shutdown. If there is a power outage, these are the backup targets for the entire cache space. In the case of a planned system shutdown, these are the backup targets for just the Control Memory region on the first two DCA boards installed in a system. During a power outage, the on-board battery power keeps the DIMMs, the SSD(s) and the board logic functioning while the DCA microprocessor destages cache to these local SSD(s). There is enough battery power to support two or three such outages back-to-back without recharging. The base SSD is 32GB (31.5GB actual), and the second one (optional) will either be a second 32GB device or a 64GB (63GB actual) device if 64GB of RAM is installed on a board.

Data Cache

The physical cache space is a single linear address space (whether a single or dual chassis) that is laid down across all DCA boards and is segmented into three distinct regions:

- Control Memory regions (2)
- DCA Cache Directories (1 per DCA board)
- User Data Region (1 per DCA board)

The overall physical cache space (laid out across 2 to 16 DCA boards) has a lower address zone (Cache A) composed of all the Cluster-1 boards (a power domain) and an upper address zone (Cache B) of all the Cluster-2 boards. Each zone has a **Control Memory** region (up to 28GB each), 2 to 8 **Cache Directories** (500MB per DCA), and a **User Data Region** (the entire remainder). The User Data Region is managed as a global pool of **64KB segments**, with large blocks of these segments checked out from or released back to a single **Global Free List** as needed by the individual MPUs.

Control Memory

The Control Memory region of physical cache contains the following types of data:

- information about the storage system configuration details
- details regarding the Parity Groups, VDEVs and LDEVs
- tables used to manage external I/O and HUR Copy operations
- Dynamic Provision and Dynamic Tiering control information

On the USP V, all of this information was contained in dedicated Shared Memory boards (with 2 or 4 per storage system). There were some issues with concurrent access to the same control information with the USP V design in the presence of particular host workloads that saw the Shared Memory system become the bottleneck. All access to each LDEV was managed within Shared Memory, so simultaneous access over multiple host ports to the same LDEV would create table lookup latencies.

Now, reads of Control Memory information are managed by the MPUs from their local memory. The only time an MPU accesses the master copy of the Control Memory region in global cache is when updating table information, when taking over for a failed partner VSD (of the same installed feature), or after replacing a failed VSD board.

Front end Director Boards (FEDs)

Overview

The **Front End Director (FED)** board controls the interaction between the storage system and those servers attached to the host ports. It also manages the connections of external storage (**Universal Volume Manager**) or remote copy products (**Hitachi Universal Replicator** and **TrueCopy**). There are three types of FED features (a pair of boards) available: 16-port Open Fibre, 8-port FCoE, and 16-port FICON. The three types of interface options can be supported simultaneously by mixing FED features within the VSP.

There are up to 8 FED boards per single control chassis (2 or 4 more can be added if the BED features are not installed). Unlike the USP V FED board, the VSP FED board does not decode and execute host I/O commands using onboard processors. In the simplest terms, the VSP FED board Data Accelerator processor accepts and responds to host requests by packaging and routing the host requests to the MPU that manages the LDEV in question. Keep in mind that each LDEV is managed by just one MPU. The MPU determines how to process these commands, manages the metadata in Control Memory, determines what addresses in cache to use, and then sends instructions to the Data Accelerator processors in the FEDs (and BEDs) as necessary. These instructions set up the transfer of data between the FED Data Accelerator and cache.

Open Fibre 16-Port Feature

The 16-port Open Fibre feature includes two boards, each with eight 8Gbps Open Fibre ports. Each port can auto-negotiate to one of three host rates: 2Gbps, 4Gbps and 8Gbps. Each board has one **Data Accelerator** processor and two **Tachyon QE-8** four-port processors.

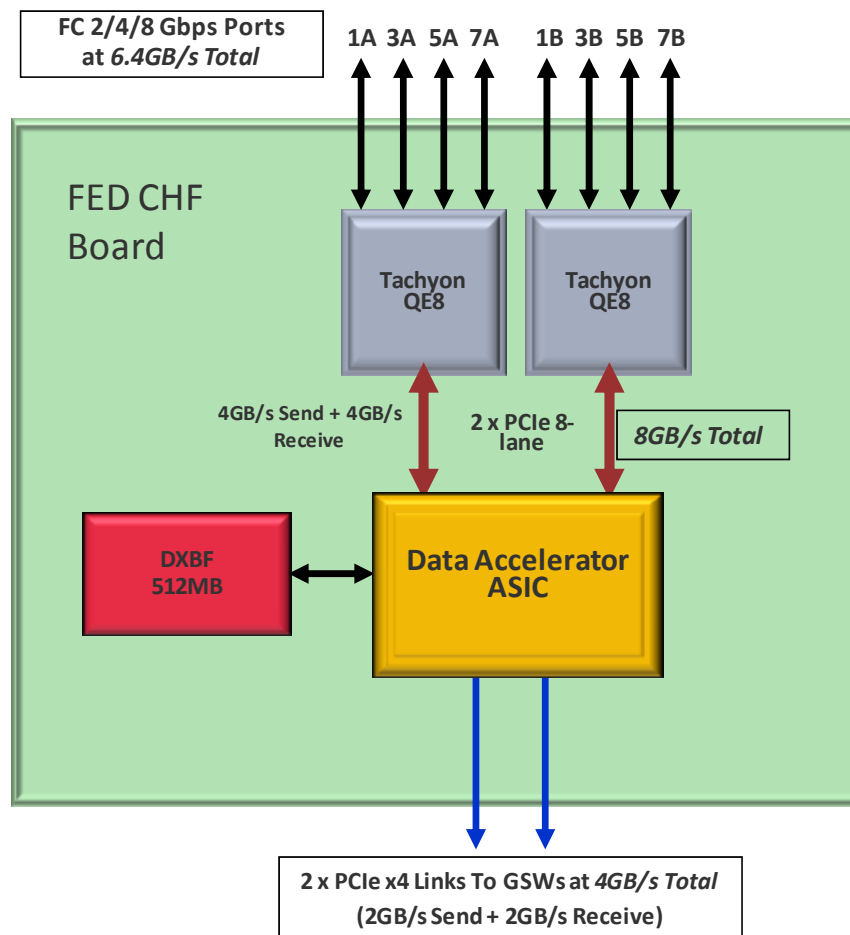
The Data Accelerator (**DA**) processor is the same as the ones used in the BED boards (described later). The Data Accelerator processor is actually two chips under one cover working together: a processor and a programmable ASIC. It contains two **Local Router (LR)** engines that communicate with the MPUs (initiating or responding to their I/O commands). The LR function uses the **local RAM** on the BED board as its work space (called the DXBF in some document) and the location of the tables it needs. The DA also contains four **DMA engines** to read or write data (or parity) directly to cache using addresses provided by the VSD MPUs. The FED DA also contains two **DRR** processors that manage RAID parity operations, data encryption, and drive formatting and rebuilds. An MPU can send DRR commands to any FC/FCoE FED board or BED board for execution, since the required data or old parity blocks will have already been loaded into cache.

The FED board local RAM (DXBF) is used for buffering host write data, maintaining LDEV mapping tables (to the managing MPUs), and housekeeping tables. When a host sends a new write to a FED port, the data is immediately buffered in the DXBF. Once the LR communicates with the owning MPU (for that LDEV), it receives a segment address in the User Data Area of cache where the blocks are to be transferred. At that point, one of the DA's DMA channels are used to duplex the write data into cache.

For reads, the MPU tells the LR where in cache to read the data, and a FED DA DMA channel is then used to directly copy it from cache to the local RAM (DXBF). At that point, the LR commands the Tachyon chip to move the data from that DXBF buffer to the host using one of the Tachyon's internal DMA channels.

There are two GSW PCIe x4 links, over which pass job requests to VSDs and user data (or RAID parity blocks) moving to or from cache.

Figure 15. 8-port Fibre Channel Board.

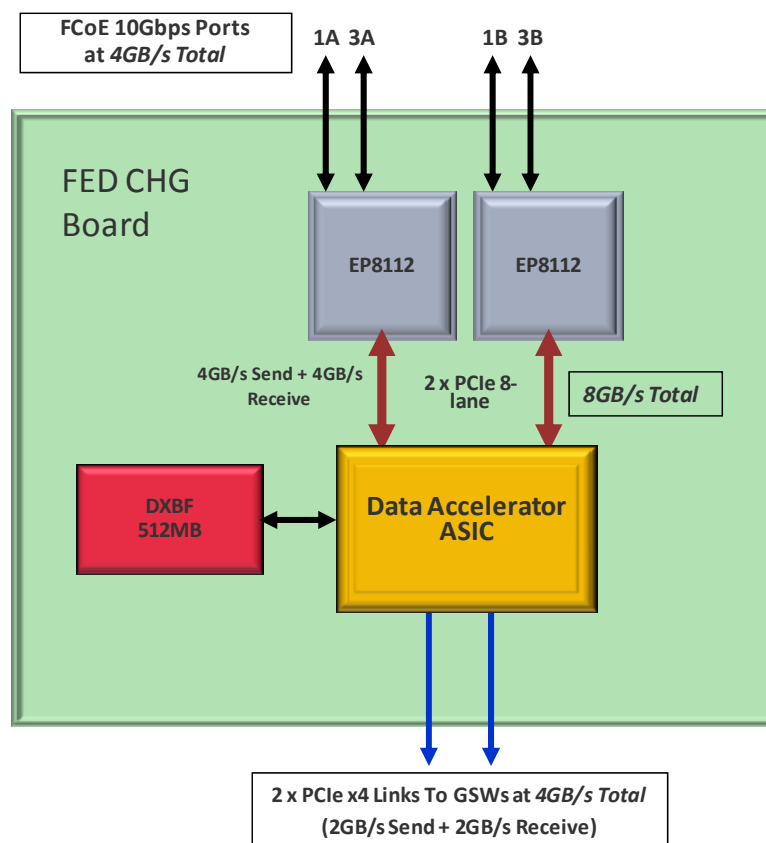


FCoE 8-port Feature

The 8-port FICON feature includes two boards with four 10Gbps iSCSI ports, where each port is fixed at this rate. Each board has one **Data Accelerator** processor and two **Qlogic EP8112 iSCSI processors** two-port processors. They can be five FCoE features (40 iSCSI ports) in a single DKC system or eleven features (88 ports) in a dual DKC system. FCoE does not support external storage, so at least one BED feature must be installed.

The Data Accelerator (**DA**) processor is the same as the one used in the FC FED or BED boards (described later). The Data Accelerator processor is actually two chips under one cover working together: a processor and a programmable ASIC. It contains two **Local Router (LR)** engines that communicate with the MPUs (initiating or responding to their I/O commands). The LR function uses the **local RAM** on the FED board as its work space (called the DXBF in some documents) and the location of the tables it needs. The DA also contains four **DMA engines** to read or write data (or parity) directly to cache using addresses provided by the VSD MPUs. The FC/FCoE FED DA also includes two **DRR** processors that manage RAID parity operations, data encryption, and drive formatting and rebuilds. An MPU can send DRR commands to any FC/FCoE FED board or any BED board for execution, since the required data or old parity blocks will have already been loaded into cache.

Figure 16. 4-port Fibre Channel over Ethernet Board



The local RAM (DXBF) is used for buffering host write data, maintaining LDEV mapping tables (to the managing MPUs), and housekeeping tables. When a host sends a new write to a FED port, the data is immediately buffered in the DXBF. Once the LR communicates with the owning MPU (for that LDEV), it receives a segment address in the User Data Area of cache where the blocks are to be transferred. At that point, one of the DA DMA channels are used to duplex the data into cache.

For reads, the MPU tells the LR where in cache to read the data, and a DA DMA channel is then used to directly copy it from cache to the local RAM (DXBF). At that point, the LR commands the iSCSI chip to move the data from that DXBF buffer to the host using one of the iSCSI chip's internal DMA channels.

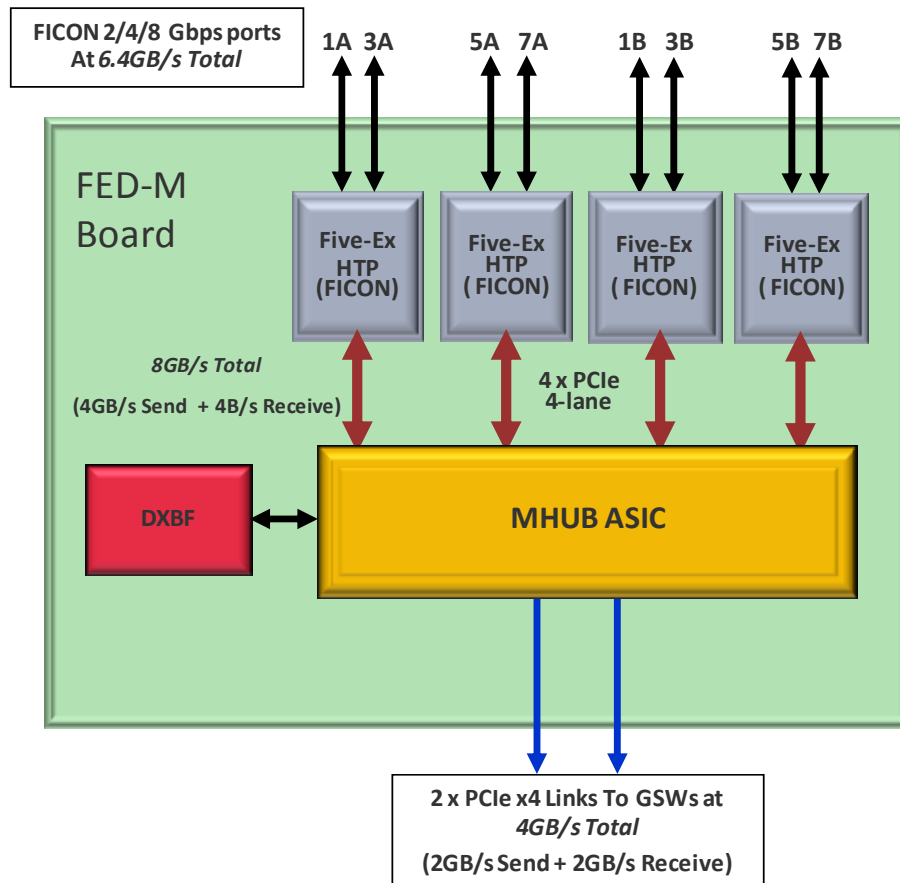
There are two GSW PCIe x4 links, over which pass job requests to MPUs and user data (or RAID parity blocks) moving to or from cache.

FICON 16-port Feature

The 16-port FICON feature includes two boards with eight 8Gbps FICON fibre ports, where each port can auto negotiate to one of three host rates: 2Gbps, 4Gbps or 8Gbps. Each board has one **MHUB Data Accelerator** processor and four **Five-Ex HTP** processors. The MHUB processor communicates with the MPUs, accepting or responding to the host I/O commands using two **Local Routers** (LR). It also has 16 DMA engines to directly read and write to cache. The MHUB ASIC is quite different from the DA processor used on the Open Fibre or FCoE boards. The MHUB is also two chips under one cover working together: a processor and a programmable ASIC. The local DRAM (DXBF) is used for buffering host data, maintaining LDEV mapping tables (to the managing MPUs), and housekeeping tables. The MHUB also

includes two **DRR** processors that manage RAID parity operations, data encryption, and drive formatting and rebuilds. There are two GSW PCIe x4 links, over which pass job requests to MPUs and user data moving to or from cache.

Figure 17. 8-port FICON Board



Back end Director Boards (BEDs)

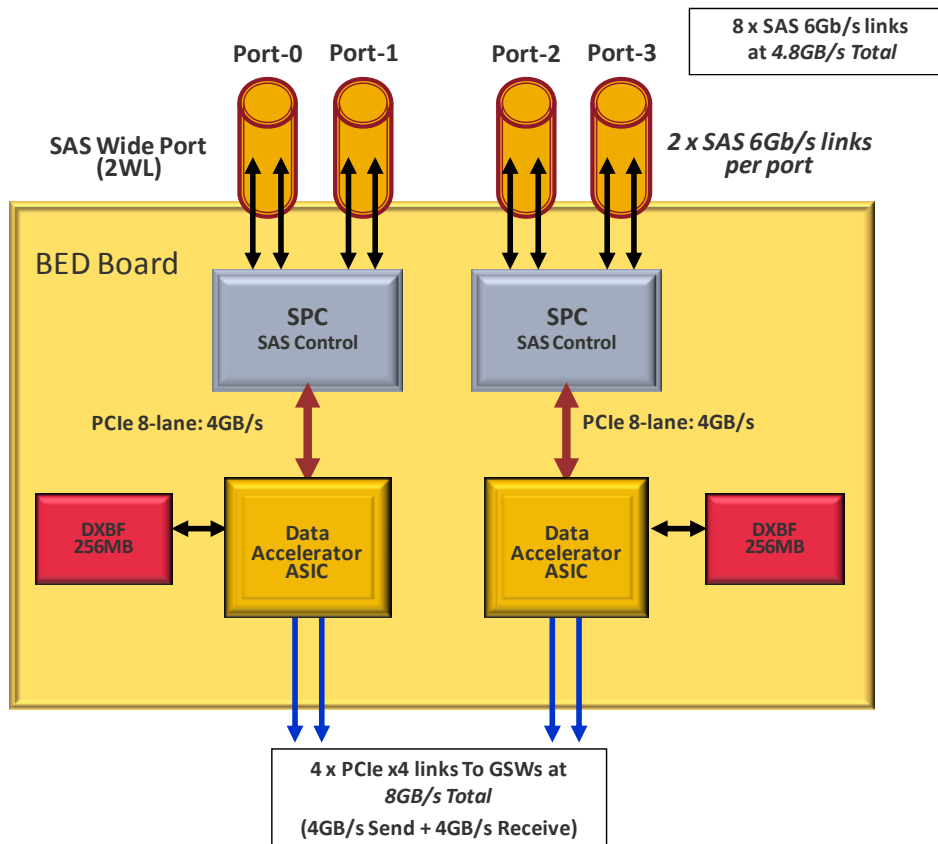
Overview

The **Back end Director (BED)** are disk controller boards that execute all I/O jobs received from the MPU processors and control all physical reads or writes to drives via the two SAS controller chips. There are one or two features (two or four BED boards) per control chassis. A single feature per chassis is called "Standard" and the dual feature is called "High Performance". Note that BEDs do not understand the concept of an LDEV, but only operate on a drive ID, a block address on the drive, and a User Data Area cache address. BED functions include the following:

- execute jobs received from any MPU
- use DMA to move data in or out of cache
- creates RAID-5 and RAID-6 parity with four embedded DRR XOR processors
- encrypt data on drive (if desired)
- manage all block reads or writes to the attached drives

Each BED board has 8 x 6Gbps full duplex SAS links. There can be up to 640 LFF drives or 1024 SFF drives per single control chassis (including up to 96 FMD drives in these totals). The DKUs and/or FBXs are attached to the 16 (Standard) or 32 (High Performance) 6Gbps SAS links available from these two or four BED boards. If this storage system is a dual control chassis configuration, then each control chassis has two or four BED boards and 1-8 DKUs, providing an aggregate of 32 or 64 SAS links for up to 2048 drives. The BEDs in each chassis only control those DKUs and drives within that chassis. But any MPU in a dual chassis storage system may send I/O jobs to any BED board in the system.

Figure 18. 8-port BED Board



Each BED board has two **Data Accelerator** processors, two **SPC SAS I/O Controller** processors, and two banks of local DRAM (DXBF). As such, there is a lot of processing power on each BED board. Each BED board provides four **2Wide SAS ports**. Each port accepts a SAS Wide cable that contains two independent 6Gbps SAS links. Each cable connects to one HDU SAS switch port on the first DKU or FBX drive container in that chassis.

The local DRAM (DXBF) is used for buffering data that is moved in or out of cache, the workspace for job requests from the MPUs, and to maintain various housekeeping tables. There are four GSW PCIe x4 links used to communicate with the MPUs and to transfer user data to or from cache.

Data Accelerator

The Data Accelerator (**DA**) processors are the same as the ones used in the FED boards. A Data Accelerator processor is actually two chips under one cover working together: a processor and a programmable ASIC. The DA contains a **Local Router (LR)** engine that communicates with the MPUs (accepting and responding to their I/O commands) and passes commands to the companion SPC SAS

controller. The LR function uses the local RAM on the BED board as its work space (DXBF) and the location of the tables it needs. The DA contains a **DMA engine** to read or write data (or parity) directly to cache. The DA also contains two **DRR** processors that manage RAID parity operations, data encryption, and drive formatting and rebuilds. Note the naming convention of the SPC 2W ports (Ports 0 to 3) – these will be used later in the SAS Engine discussion.

SPC Processor

Each SPC SAS processor contains a high performance dual-core CPU that has the ability to directly control SAS or SATA devices (using SAS “link protocol encapsulation”) over its four full duplex 6Gbps SAS links. The SPC creates a direct connection with one drive per link via the drive box HDU switches when the SPC needs to communicate with a drive. There are no loops as with Fibre Channel. The speed at which a link is driven (3Gbps or 6Gbps) depends upon the interface speed of the selected drive for that individual I/O operation. Those drives with the 3Gbps interface are driven at that rate by the SAS link, and those drives that are 6Gbps are driven at that higher rate whenever a BED SAS link is communicating with them. All drives currently available on the VSP are of the 6Gbps variety.

SAS Engine Concepts

The VSP uses a **SAS Engine** back end architecture that is very different from the one used by the earlier generation Fibre Channel based USP and USP V systems. The SAS Engine is my name for the collection of components used in the back end. Each control chassis has a separate SAS Engine which contains the following elements:

- 2 or 4 BED boards, each with dual Data Accelerator and SPC processors
- 8 or 16 SAS 2W cables (16 or 32 6Gbps SAS links)
- 1 to 8 DKUs, the frame mounted drive containers (as UBX-LFF or SBX-SFF versions) with 8 HDUs
 - 8 to 64 HDUs with dual SAS switches, each with slots for 10 (UBX) or 16 (SBX) drives
 - 640 to 1280 Drives in the HDUs (SSD, SAS, or SATA)
- 1 or 2 FBXs, the frame mounted Accelerated Flash containers (reduces the DKU count)
 - 1 to 4 FMUs with dual SAS switches, each with slots for 12 FMDs
 - 48-96 FMD devices in the FMUs

Refer to Figures 19 and 20 (BED SAS links to DKUs/HDUs) for much of the following discussion. **Note that the various factory manuals nearly always show paths as 2W cables, not as SAS links (two per cable). Since they are rarely labeled as to which kind they are, this creates some confusion on this subject.**

BED SAS Links to FBX/DKU HDU Overview

Figures 19 and 20 provide views of the BED-to-HDU associations. This is an important diagram to understand as it is one of the many differences with the previous USP V design. The USP V uses a Fibre Channel Arbitrated Loop backend that is connected using FC switches in the HDUs. The VSP BED SPC processors manage certain HDUs from the FBXs/DKUs within a chassis. The HDUs are organized into **SAS Boundaries** (or “HDU stacks”), having the same port from each SPC connected to them. For example, in the first stack (starting with DKU-0, HDU-00-0, then HDU-00-4, and so on), all drives in this stack will be driven by the Port-0’s on SPC-A, SPC-C, SPC-E and SPC-G (these are illustrative labels only). The VSP uses four or eight 6Gbps SAS links per SAS Boundary (what I call an HDU “stack”), with the switches able to

connect to four or eight drives within that HDU set at the same time, or to pass directly through to the next HDU switch in that HDU stack.

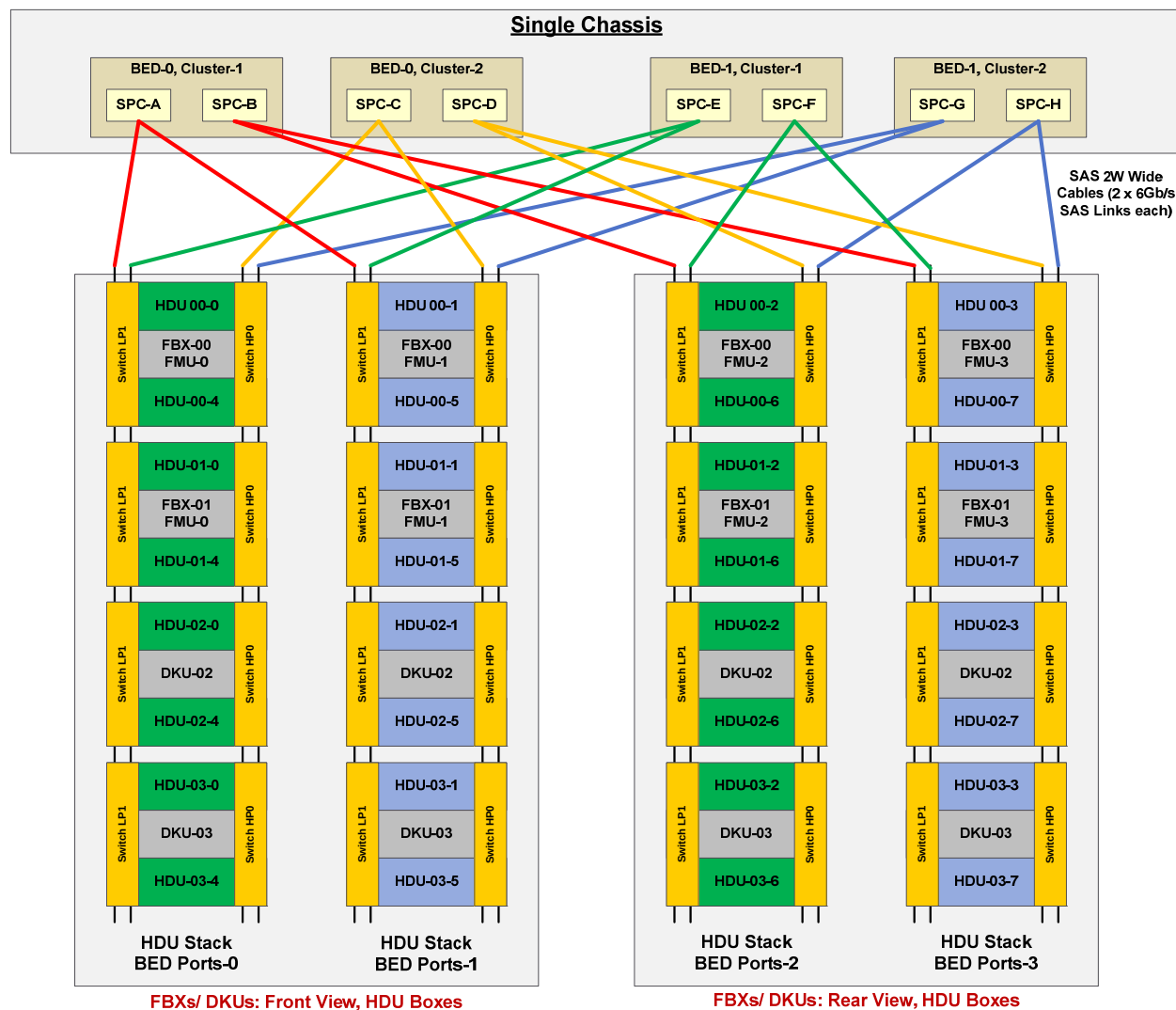
Figure 19. Back end SAS Link Layout (Standard Performance Configuration, 8 2W cables, 16 SAS links)



Figure 19 above shows the single chassis **Standard Performance** configuration (basic BED feature) with four DKUs (showing the front and back views) and 32 HDUs installed in one chassis. This provides 16 6Gbps SAS links (eight 2W cables) that attach to a maximum of 640 LFF drives or 1280 SFF drives (or some mixture of those). Each DKU has 16 active 6Gbps SAS links passing through it, with two links attached to each expander switch. This BED-to-DKU association is fixed within each chassis. A dual chassis storage system would have two of these independent structures (no cross connections) with up to 1280 LFF drives or 2048 SFF drives (or some mixture) in all.

Figure 20 shows the single chassis **High Performance** configuration (two BED features) with two FBXs and two DKUs. This provides 32x 6Gbps SAS links (16x 2W cables) to up to 96 FMD devices, and 480 LFF or 768 SFF drives (or some mixture of them) if using six DKUs. Each DKU/FBX has 32 active 6Gbps SAS links passing through it, with 8 per SAS Boundary (the “HDU stack”). A dual chassis storage system would have two of these structures supporting (in this same configuration) 192 FMDs and 960 LFF drives or 1536 SFF drives (or some mixture).

Figure 20. Back end SAS Link Layout (High Performance Configuration, 16 2W cables, 32 SAS links)



The four Accelerated Flash FMU units in the first FBX-00 (FBX-10 in the second chassis) are directly attached to the sixteen SAS 2W cables from the two BED features. These are FMUs 00-0, 00-1, 00-2, and 00-3. [For the second VSP chassis, these would be 10-0, 10-1, 10-2, and 10-3.] The remaining HDUs in that single chassis (in each “stack”) are attached in a daisy-chain to these four base HDUs.

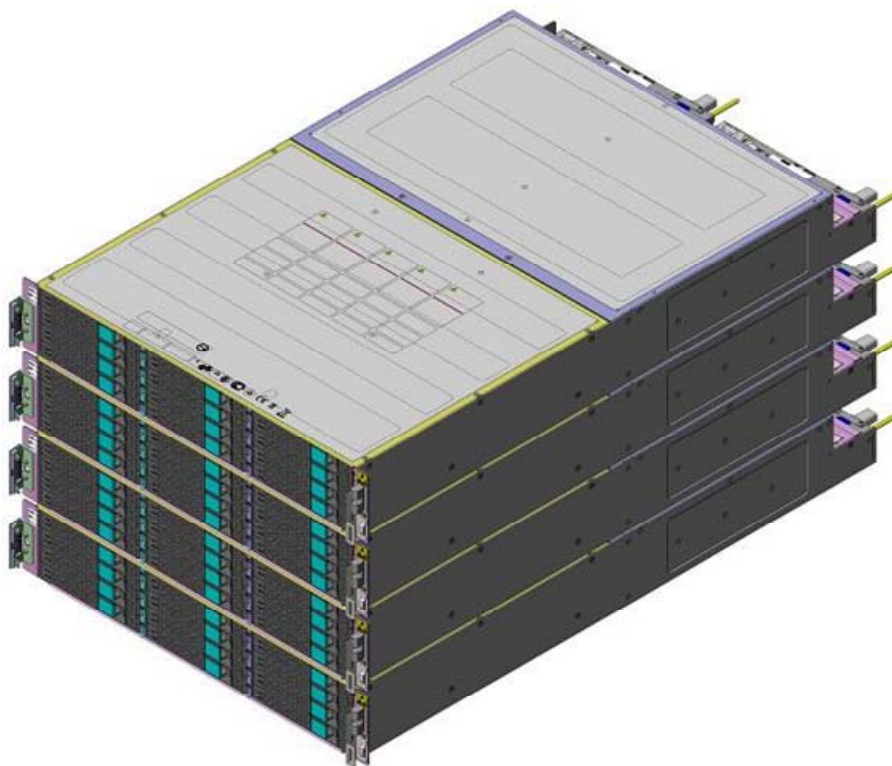
Note that in the dual chassis configuration, while any FED board can interact with any MPU or DCA cache board, and any MPU can interact with any BED board, there is an exclusive association among the BED boards and the FBXs or DKUs by chassis. There is no cross-chassis connection at this level since that wouldn’t be possible.

FBX Details

The FBX is the disk container for the new Hitachi custom built Accelerated Flash drives. The FBX is far simpler than a DKU. It is basically a 13U rack mounted box with sliders for up to four of the 2U FMU “trays” installed. The 13U space requirement includes the following:

- A 2U space left at the top for cables
- An 8U space for the four FMU trays
- A 2U space under the FMU space for cables
- A 1U space for alignment to match the 13U spec for the DKUs

Figure 21. View of the FBX and four FMUs.



Each FMU tray contains two power supplies (Cluster-1 and Cluster-2), two ENC SAS switches, and two HDUs with up to six FMD drives each. Each DKU drive box (either SBX or UBX) contains eight HDU disk boxes. The FBX has been designed so that each of its four FMU trays has two six-drive HDUs so that the FBX also presents eight HDUs to the system.

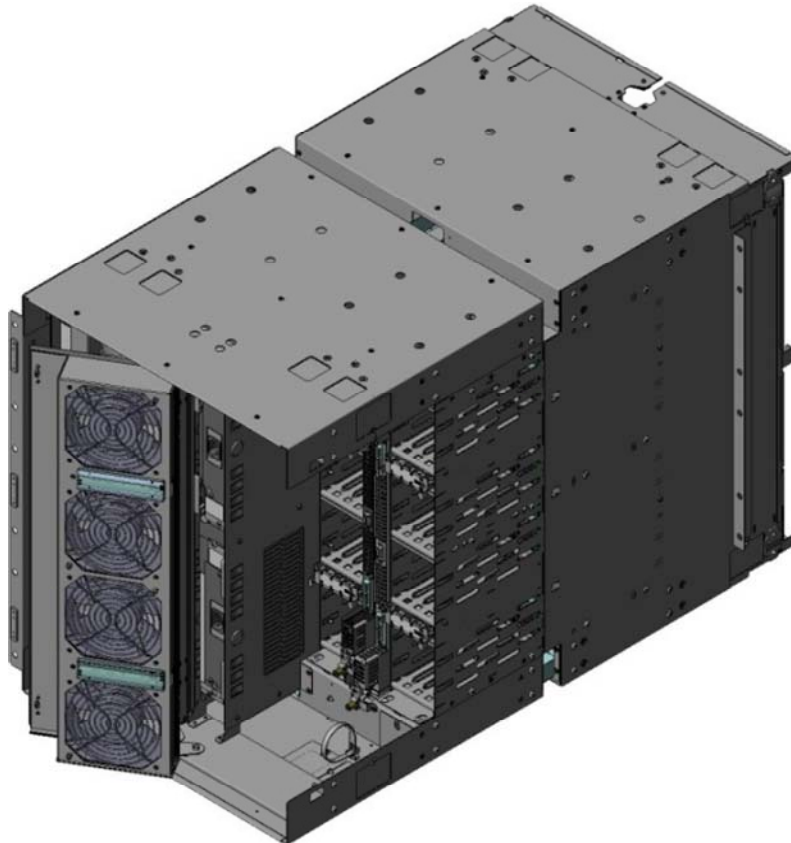
In the VSP, peak performance from the FMD LDEVs is achieved with a dual chassis configuration, where each chassis has one fully populated FBX in the first position of the HDU stacks and uses the optional Flash Acceleration feature (software).

DKU Details

Shown below in Figure 22 is a view of a DKU box. There are two fan doors on the front and two more on the rear. One fan door is open in this picture. When a fan door on the front or rear (an interlock prevents opening both at the same time) is opened, those fans stop. The fans on the closed door at the

other end are then run at double speed to maintain the same airflow. When these doors are all closed, the fans run at half speed (much quieter).

Figure 22. Illustration of a DKU Box



A DKU is organized into a front and a rear power zone (Cluster-1 and Cluster-2), each with four HDU modules (eight total). The LFF DKU (UBX) feature holds 80 drives (10 per HDU) while the SFF DKU (SBX) feature holds 128 drives (16 per HDU).

Figures 23 and 24 illustrate the four HDUs and their drive slots in a DKU as seen from the front side. An empty DKU weighs about 100 pounds, so it may make sense to pre-install some empty DKUs on a new system to save on labor costs later on. Note that within each DKU there is a pair of HDUs associated with a pair of SAS Switches.

Figure 23. HDU and Drive Slot Layout of the LFF DKU (front side, Cluster-1)

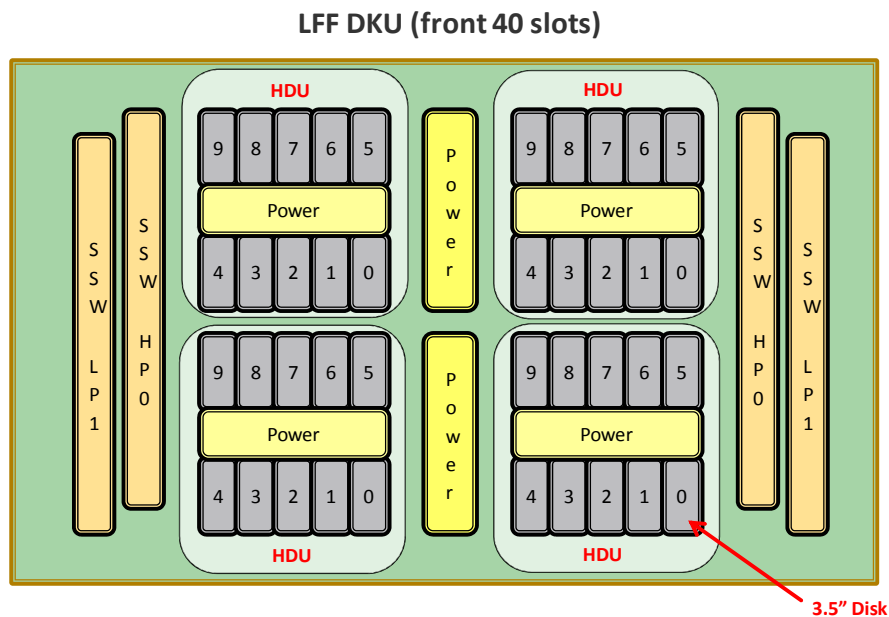


Figure 24. HDU and Drive Slot Layout of the SFF DKU (front side, Cluster-1)

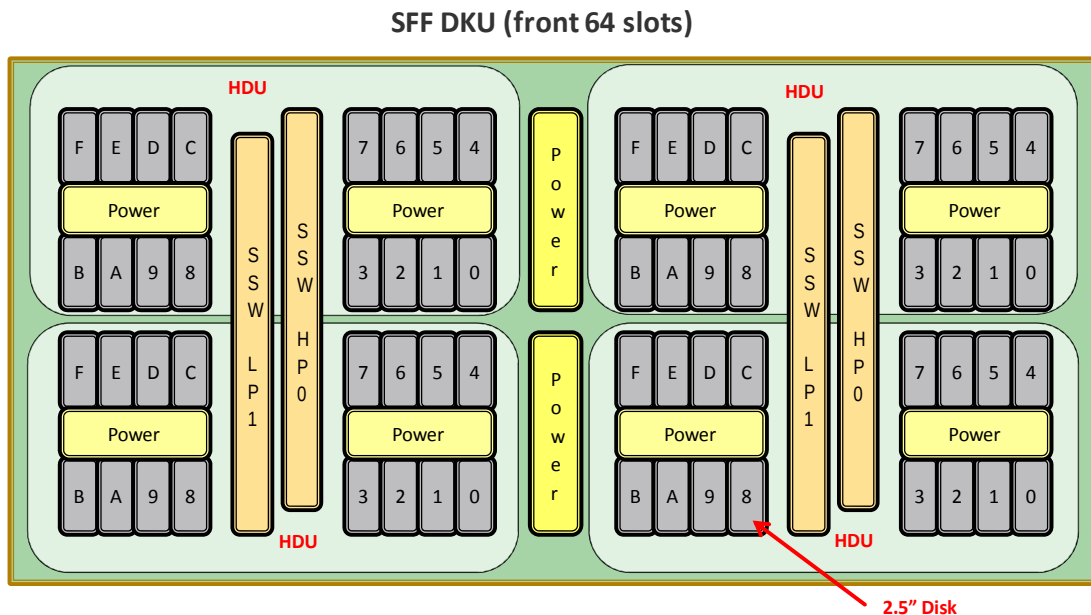


Figure 25 shows how the SAS links pass through the four pairs of HDUs per DKU. Notice how there is a vertical alignment within the DKU by an HDU pair, where those two HDUs share a common pair of SAS switches. Switch HPO is attached to the one or two BED boards in power Cluster-1 and switch LP1 attaches to the one or two Cluster-2 BED boards. Each such pair of HDUs and their dedicated SAS links (4 or 8) is known as a **SAS Boundary**. Each colored arrow indicates one (single BED) or two (dual BED) SAS 2W cables (one from each BED feature) where each 2W cable has two active SAS links (refer back to Figure 20 to follow this).

So, on a VSP single control chassis with only one BED feature installed, there are 16 SAS links passing through that chassis' stack of DKUs (1 to 8 of them), with 4 SAS links passing through each DKU's stack of HDUs (SAS Boundary). If there are two BED features, then there are 32 SAS links with 8 SAS links passing through each stack. See Appendix 7 for some explanation of the DKU diagrams in the Maintenance Manual that create a degree of confusion on this detail.

Figure 25. SAS Boundaries

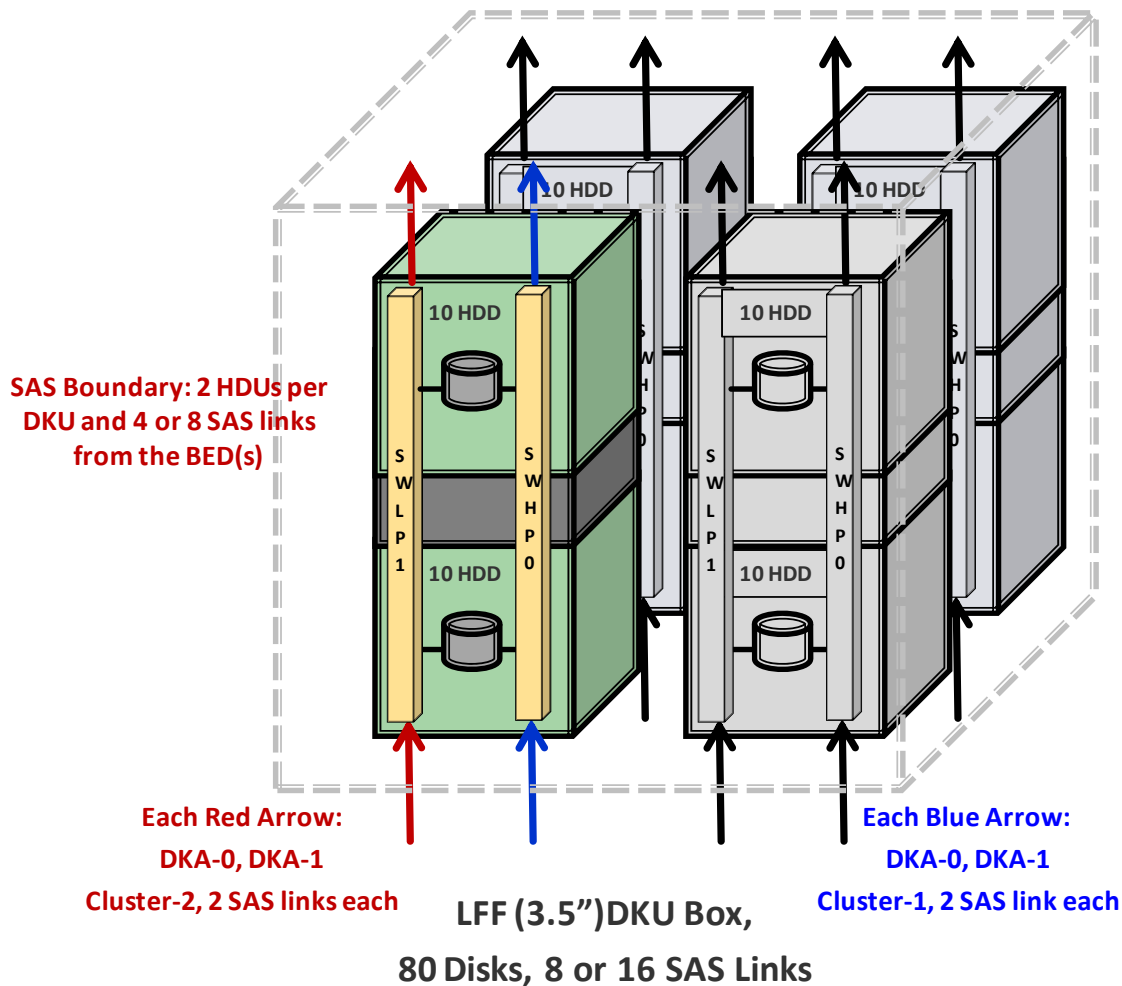
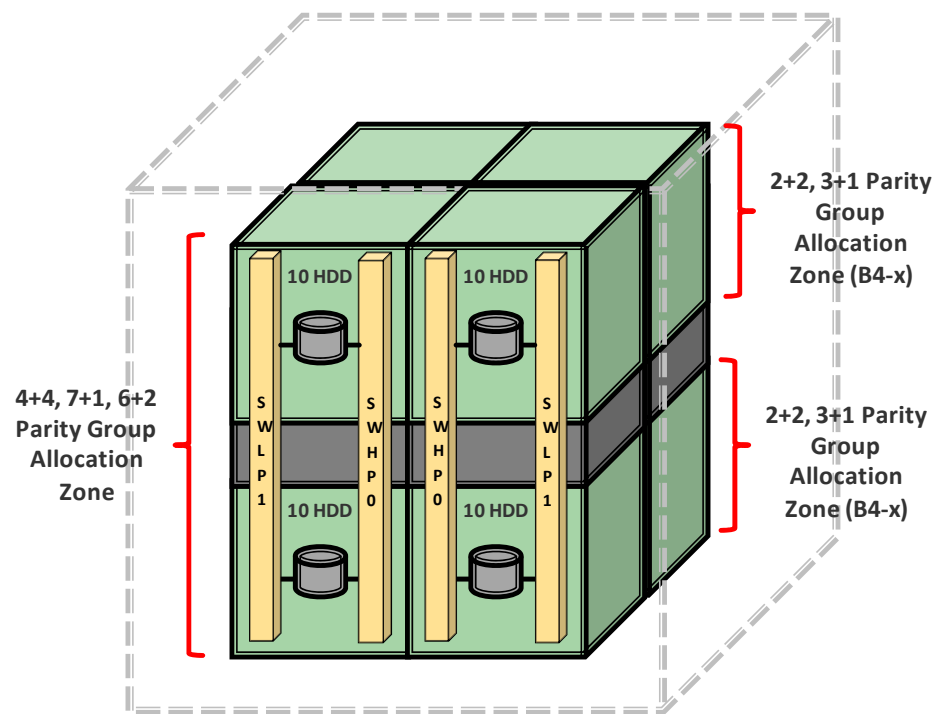


Figure 26 indicates the organization of "B4-xx" zones within a DKU into which Parity Groups are defined. Each upper horizontal "slice" of four HDUs and each lower slice of four HDUs in a DKU is a specific "B4-x" zone. Parity Groups get their names from the "B4-x-y" style, where "x" is the zone and "y" is the Parity Group number. For four-disk Parity Groups (RAID-5 3D+1P, RAID-10 2D+2D), all drive members come from a single B4-x zone. For the eight-disk Parity Groups (RAID-5 7D+1P, RAID-10 4D+4D, RAID-6 6D+2P), the eight drives come from the two B4 zones within the same DKU, but they take their name from the lower zone.

Notice how these drive zones cross the SAS links and power domains (clusters). Appendix 6 shows the B4 names for all 16 DKUs in a dual chassis system. Note that an individual Parity Group (whether 4-disk or 8-disk) will have all available SAS links in that same control chassis driving its members.

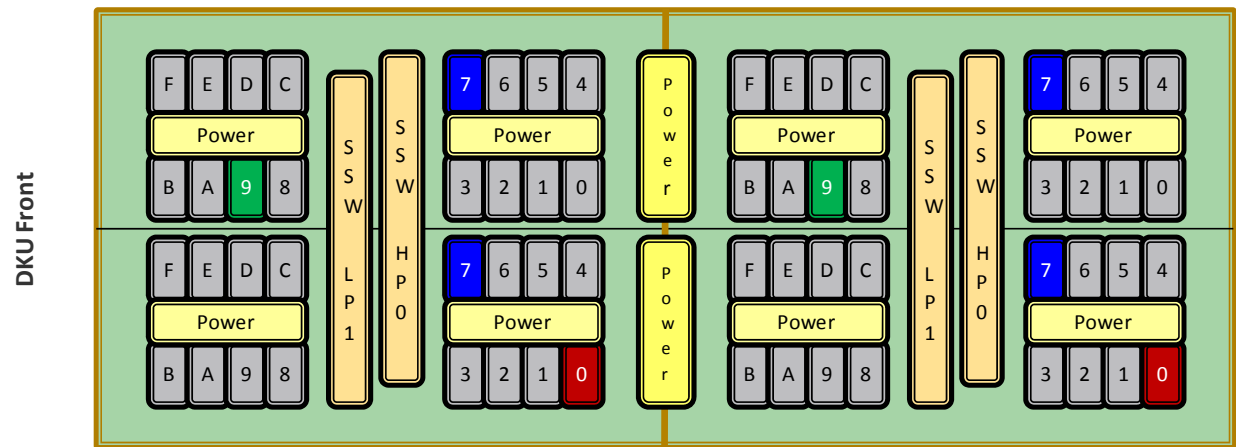
Figure 26. Organization of a DKU into Horizontal Parity Group Zones (“B4s”)

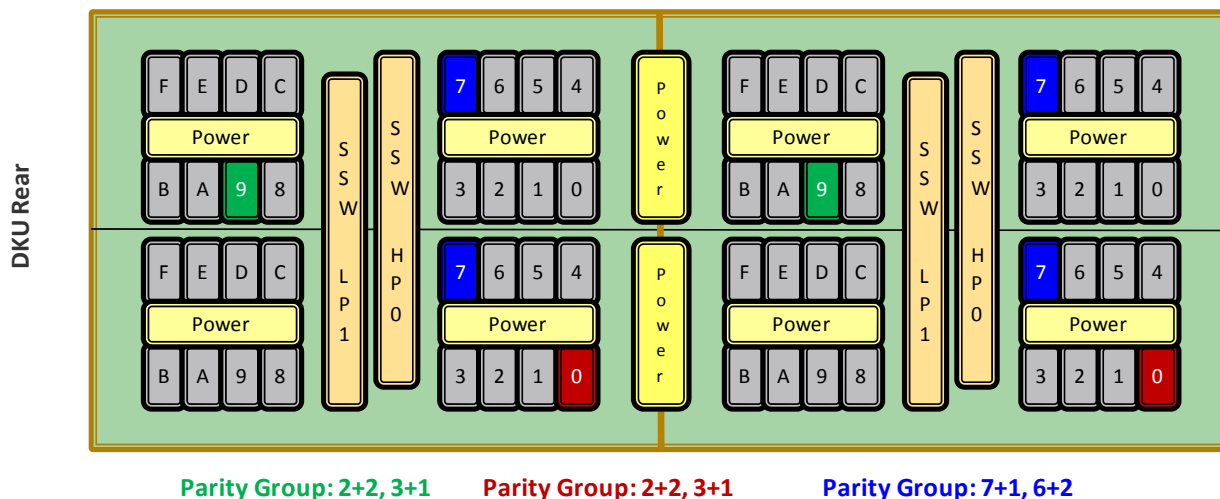


LFF (3.5 inch) DKU Box

Figure 27 illustrates the location of drives per Parity Group using an alternate style front and rear view of one DKU and its 8 HDUs. The green or red drives form 4-disk parity groups (such as 2+2, 3+1), while the blue ones show a native 8-disk Parity Group (such as 7+1 or 6+2). There is also a native 16 drive Parity Group now as a 14+2.

Figure 27. Alternate View of the Location of drives per Parity Group in an SFF HDU

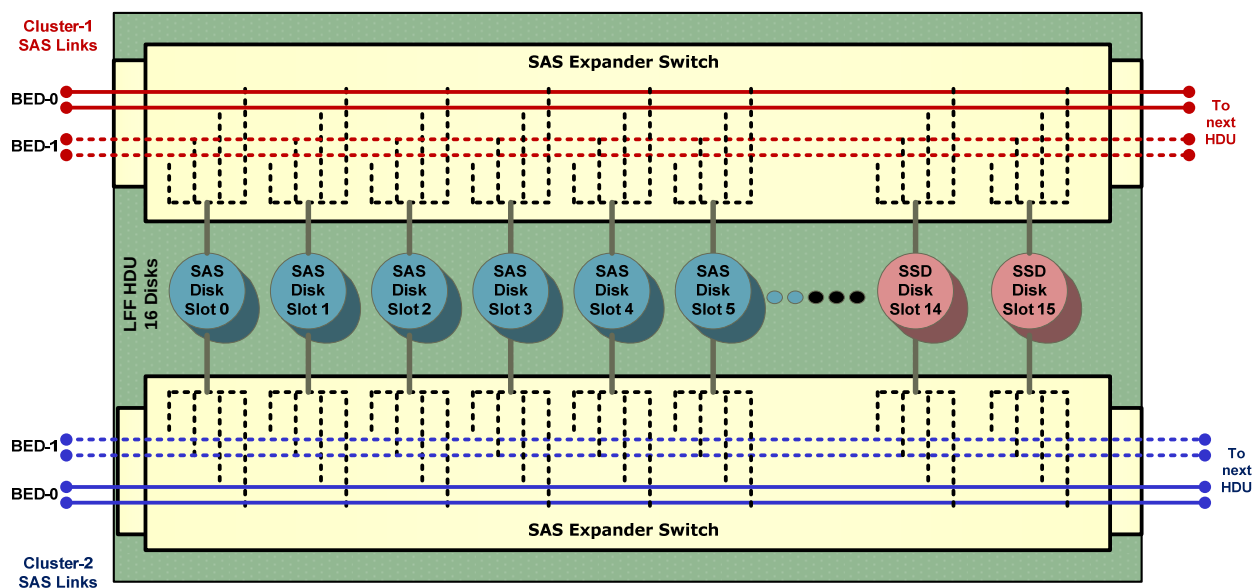




DKU- HDU Internal Details

Each HDU is attached to two internal SAS switches within the DKU. A pair of HDUs in each DKU shares these two switches. The switch named **LP1** is attached to the Cluster-2 BED ports, and the one named **HPO** is attached to the Cluster-1 ports. Recall that each port provides two independent SAS links. This means that each LP1 will either have two active SAS links on a chassis with one BED feature or four active links if there are two BED features. Figure 28 provides an illustration of the SFF HDU. Each SPC processor determines which of its two links per port (via the 2W cable) will be used to connect to each drive.

Figure 28. View of the SAS Links, Switches and Drives in an SFF HDU



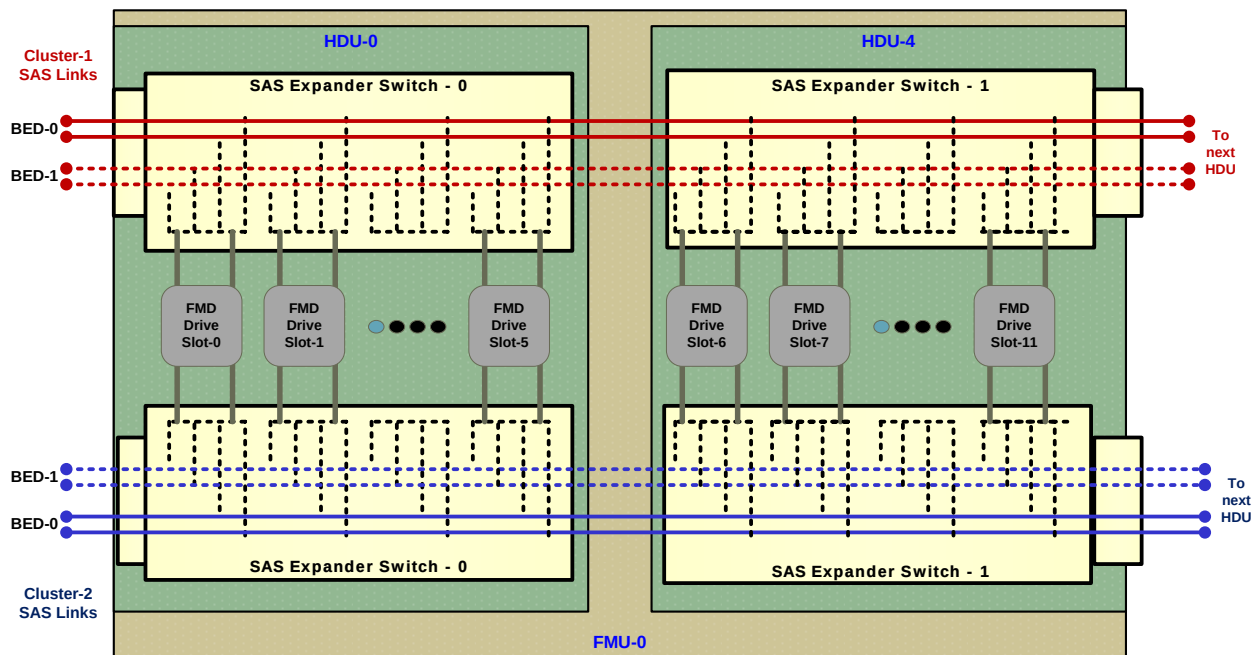
All drives within each HDU are dual attached with one full duplex drive port going to each switch. Each SATA disk has a special drive canister with a logic board to make a single half-duplex ported SATA disk *appear* to have two ports.

FBX- HDU Internal Details

Each FMU tray contains two separate HDUs. For FMU-0, the two HDUs are named HDU-0 and HDU-4. For FMU-1, they are HDU-1 and HDU-5, and so forth. Each HDU has two SAS switches. The switches named **LP1** are attached to the Cluster-2 BED ports, and the ones named **HPO** are attached to the Cluster-1 ports. Recall that each BED port provides two independent SAS links. This means that each LP1 will have eight active SAS links if there are two BED features per chassis (recommended) when configuring FBXs.

Figure 29 provides an illustration of an FMU tray with its two HDUs. Each BED SPC processor determines which of its two links per port (via the 2W cable) will be used to connect to each FMD drive. Note that FMD drives have four ports rather than two as with SAS drives. The FMD drives are installed as a matrix of 4 high by 3 wide in the FMU tray.

Figure 29. FMU Tray HDU Details



Disk Details

Installable Drive Features

The single control chassis VSP storage system supports a mixture of up to 96 FMD drives, 1024 SFF drives (SSD, SAS) or 640 LFF drives (SSD, SATA) or some value in between depending on the mix of FBXs and DKUs installed in a chassis. The dual control chassis storage system supports twice as many BEDs,

FBXs, DKUs and drives. Each FBX can hold 48 drives, each SBX DKU can hold 128 SFF drives and each UBX DKU can hold 80 LFF drives. There may be up to two FBXs and six DKUs (eight without FBXs) per control chassis, so the total drive counts will be determined by how many FBXs/DKUs are installed and of which type of DKU.

The single chassis VSP storage system has an overall limit of 128 SSD drives, with up to 256 SSDs possible in the dual chassis storage system configuration. **Note: If the SSD Parity Groups have sustained, high thread count random workloads driven against them, then 32 SSDs per chassis should be considered the usable limit, where the four VSD boards will likely be saturated.**

Table 2 illustrates the advertised size (decimal) and the raw size (binary like RAM) after Parity Group formatting of each type of drive. Also shown is the “rule of thumb” average expected random IOPS rate for each type of drive. **Decimal** means that 1GB is shown as 1,000,000,000 bytes. **Binary** (the number system used in computers and memory) would show 1GB as 1,073,741,824 bytes. On most computers, a 1GB drive would show up as less than 953.7MB just due to this difference in number systems. The 953.7MB value is what most operating systems would report as the size of a raw unformatted device. File system formatting usually cuts this by about 4-5%, so 953.7MB might be about 916MB usable to applications.

Table 2. Table of Drive Types and Rule-of-Thumb IOPS Rates

Drive Type	RPM	Form Factor	Port Speed	Advertised Size (GB)	Raw Size (GB)	Nominal Random Read IOPS
1.6TB FMD	Hitachi MLC	FMD	6 Gbps	1600	1759	20,000+
3 TB SAS	7200	LFF	6Gbps	3000	2937	100
2 TB SATA	7200	LFF	6Gbps	2000	1969	80
900 GB SAS	10k	SFF	6Gbps	900	864	140
600 GB SAS	10k	SFF	6Gbps	600	576	140
400 GB SSD	HGST MLC	SFF	6Gbps	400	393	5000+
300GB SAS	15k	SFF	6Gbps	300	288	200
300 GB SAS	10k	SFF	6Gbps	300	288	140
200 GB SSD	-	SFF	6Gbps	200	196	5000+
146 GB SAS	15k	SFF	6Gbps	146	137	200

The type and quantity of drives used in a VSP and the RAID levels chosen for those drives will vary according to an analysis of the customer workload mix, cost, application performance targets, the usable protected capacity requirements, and the rebuild time for a failed drive.

All LDEVs (actually pointers to VDEV containers – see Appendix 4) that are created from Parity Groups that use SATA drives can be configured either as OPEN-V or 3390-M emulation; therefore these internal SATA VDEVs are usable for mainframe volumes.

FMD Drives

Hitachi’s custom FMD drives could be described as turbo-charged SSDs. Each FMD has much higher performance than an SSD due to much higher internal processing power and degree of parallelism. The

first drives are 1.6TB, with 3.2TB drives planned for later release. Inside each FMD is a 4-core processor and ASIC (the Advanced Switch Flash Controller - **ASFC**) that controls the internal operations, the four external 6Gbps SAS ports, and the 128 flash memory chips. An SSD drive has one processor and 16 or 32 such flash memory chips. There is a considerable amount of logic in the ASFC that manages the space and how writes are managed. This greatly aids in substantially increasing the write rate for an FMD drive over a standard SSD drive.

SSD Drives

SSD drives are a fairly new technology in storage systems. “Drive” is the proper term since these are not “disks” in any sense. SSD drives are very costly in both the “\$ per GB” metric and the “\$ per IOPS” metric, but they do provide much lower read miss response times – the primary reason for using SSDs. SSD drives are currently available in two sizes: 200GB SFF and a 400GB SFF models.

SSDs are often going to be used with RAID-5 (7D+1P) for read intensive workloads and RAID-10 (2D+2D) for write intensive workloads. For environments that must avoid dual drive failures at all costs, then RAID-6 must be used, as SSD drives do not have higher reliability than do regular disks. Although there are no seek times or rotational delays on SSDs, there is a substantial difference between the use of these three RAID levels with regards to the number of system cache blocks in use (data and parity) and the extra back end read and write drive operations on the BEDs. Although these can be sub-millisecond operations, compared to the access times of 5-7ms for SAS drives and perhaps 14ms for SATA drives, the RAID operations still add to the overall BED loads.

It is with small block random read workloads where SSDs justify their high costs. Keep in mind that the MPU microprocessor IOPS capability is the same for FMDs, SSDs and HDDs. The storage system IOPS **capacity** doesn’t increase with SSDs, even if that same IOPS can be obtained using far fewer SSDs.

SSDs are less cost effective for large block sequential transfers compared to HDDs. Large block sequential transfers tend to be limited by cache switch path bandwidth data transfer rate in MB/s rather than MPU processor IOPS capability.

SAS Drives

SAS drives are the primary drives used now in both enterprise and midrange storage systems from Hitachi. These are the workhorse drives, having both high performance and good capacities. They fall in between the random performance levels of SSD and SATA (much closer to the SATA end in comparison to SSD rates). All three drive types will have about the same sequential read performance per matching RAID levels. SAS disks are the same as Fibre Channel disks but with a SAS interface. SAS drives used in the VSP come in 15K RPM, 10K RPM, and 7.2K RPM rotation speeds, and in sizes currently ranging from 146GB to 3TB. The SAS interface speeds on all Hitachi supported drives is 6Gbps.

SAS drives are designed for high performance, having dual host ports and large caches. The dual host ports allow four concurrent interactions with the attached BEDs. Both ports can send and receive data in full duplex at the same time, communicating with the DRAM buffer in the drive. However, only one data transfer at a time can take place internally between the DRAM buffer in the drive and the read/write heads.

SAS drives may be low level formatted to native sector sizes of 512 bytes and 520 bytes. Hitachi uses the 520 byte format. All host I/Os to the VSP storage system are done using the usual 512-byte sectors. The storage system adds 8 check bytes to each host 512-byte sector at the point where the data arrives at the storage system port from the host. These 8 check bytes have nothing to do with mechanisms internally within the drive. Instead, the 8 check bytes are used to protect against potential errors in internal storage system processing.

The check bytes include two things. An encoding of the LDEV name and LBA within the LDEV are included in the check bytes to ensure that there were no errors in the virtualization mechanisms in the storage system. When reading from drive, if the check bytes match you know that the sector being read from drive was originally received from the host at the same offset within the same LDEV as the sector the host is requesting to read. In other words, the encoding of the LDEV name and LBA in the check bytes makes sure you don't make any mistakes in where you put the sector when writing it and don't make any mistakes in figuring out where to read the data from drive.

The check bytes also include a checksum over the sector contents. This checksum is computed at the time the data was originally received from the host, and is stored in the check bytes along with the data. This is used to implement end-to-end data integrity protection within the storage system. The 520-byte sector is transferred from the FED to cache to drive to cache to FED, passing over various busses and being manipulated by various microprocessors and other hardware logic. If the sector payload checksum still matches when the 512-byte host sector is being transferred to the host, then you know that the storage system did not corrupt the data internally.

It is beyond the scope of this document to talk about drive internals. (If you are interested, see the Ian Vogelesang blog posting on this topic. Navigate to <http://techops.hds.com/>, and then click on Blogs and see the archive from July, 2010.) The drive internally has the same challenges and meets them with analogous mechanisms.

The first SAS drives were of the usual Large Form Factor size (3.5"). Now they have shifted to the smaller Small Form Factor (2.5"). This reduces cost and heat, as well as reducing seek times (smaller platters). A 2.5" SFF 10K RPM SAS drive will have higher random performance than a 3.5" LFF 10K RPM FC drive, but lower performance than a 3.5" LFF 15K RPM FC drive for 100% read loads. Also, the SAS 7200 RPM drives have essentially replaced the use of much slower and less capable SATA drives.

SATA Drives

SATA drives offer very high capacities (now at 3TB per drive) at an "economy" level of performance. These are best suited for archival duty. These are not suitable for high levels of random workloads (like OLTP) with even modest (10% or so) sustained levels of write.

Due to the large capacities, most SATA drives are used in RAID-6 configurations in order to avoid the likelihood of a dual drive failure during the typically long rebuild times of a failed drive onto a spare drive. However, the use of RAID-6 carries a very high RAID write penalty factor of six internal storage system drive operations per host write request. With SATA, there are three more such operations (read-verify) per write, or nine in all per host write request. Some customers will trade off usable capacity for a large increase in write performance by the use of RAID-10 rather than RAID-6. RAID-10 only carries a write penalty of two drive operations per small-block host write request. The usable capacity is reduced from 80% (RAID-6 6D+2P) to 50% (RAID-10 4D+4D). But the cost per usable gigabyte is not as important as the cost per IOPS for many customers.

The SATA drive canister has a logic board that creates two SAS ports from the single SATA port on each SATA drive. For SATA-W/V type parity groups, this interface also performs a read-after-write check on the last sector from the range of blocks (whether data or parity/mirror) just written to the drive. After an application block is written to the drive, the last sector of it is read from the surface and compared to the buffer in the canister logic board.

Other considerations on the use of SATA versus SAS drives when using RAID-6 include:

- the no-load format times are about 18X longer than with SAS drives (no second on-board processor to perform a format)
- the no-load corrective copy times are about 4.4X longer than with SAS drives
- the no-load drive copy time is about 10X longer than with SAS drives
- there should be 1 spare SATA drive for every 16 SATA drives in use

Hitachi's recommendation on suitable uses for SATA volumes is to use them for near-line storage with a low frequency of access but online availability, such as:

- Email archives, backups, 2nd and 3rd copies
- Historical data with limited access (medical records, etc.)
- Archived documents for regulatory purposes
- Sound or video files
- Tape replacement

The decision to use up valuable internal chassis drive slots with SATA drives rather than the much higher performing SSD or SAS drives should be carefully considered. In many cases, the use of SATA drives within virtualized storage on an external midrange product might make better sense. However, there will be individual cases where the use of some internal slots for SATA drives will solve a specific customer business solution in a straight forward manner. It is not expected that a VSP would normally be configured with a significant percentage of internal SATA drives.

SATA Drive Sector Details

SATA drives have a fixed sector size of 512 bytes. As a result, the storage system cannot write each internal 520-byte sector consisting of the 512-byte sector received from the host together with the 8-bytes of storage system protection code as a single sector on drive. Instead, the storage system writes to SATA drive a group of 64 x 520-byte internal storage system sectors as a group of 65 physical 512-byte SATA sectors.

When the host writes a small random block to an LDEV carved from a SATA parity group, the storage system must first do a "pre-read" to read the 32K group of 64 x 520-byte sectors, then update with the block received from the host, before writing to drive.

In this way, the storage system continues to be protected against internal storage system hardware or microcode errors, even with drives that cannot be formatted with 520-byte sectors. Storage system vendors that do not write storage system check bytes with each host sector cannot protect against these types of failures.

Cache and RAM Overview

The memory systems used within the VSP design are quite different than those used in the USP V design. [See the *VSP Cache Operations Theory* paper listed in the References for a deep dive into the cache system.] This is mostly due to the elimination of the USP V's discrete Shared Memory system and the dedicated I/O processors (MPs) on the FED and BED boards. The VSP now uses a dynamic, partitioned cache allocation system in order to provide for the buffer space for each VSD board processor (MPU) for use in managing their discrete set of LDEVs. Each MPU has a set of LDEVs for which it controls all I/O operations. This distribution is normally fairly uniform across the installed sets of VSD boards.

The VSP storage system has a single **Global Free List** that manages all **64KB cache segments** for the User Data Area on the cache boards. [All of cache is managed as a collection of these 64KB segments.] Each MPU also operates a **Local Free List** (of those segments allocated to it from the Global Free List) from which it privately draws and returns segments for individual I/O operations on those LDEVs it manages. There is also a fixed, shared region on the first two cache boards used by all of the VSD boards for the system's **Control Memory** master copy, plus a 500MB Cache Directory on each cache board.

Memory Organization

The VSP has three parallel **logical** memory structures that manage access to data, metadata, control tables and storage system software. These include:

- **Control Memory** for configuration data, metadata and control tables. Kept in local DRAM on each VSD board, along with a master global copy in a region of system cache.
- **VSD Cache Regions** (primary cache region for user data blocks, including the mirrored write blocks, parity blocks, and Copy Product data blocks).
- **Local workspace** on each VSD, FED and BED board for their embedded processors.

The VSP has three types of **physical** memory systems, which include:

- **Cache System** (on the Data Cache Adapters) – the storage system's primary cache space of up to 1024GB per dual chassis system (if using 8 DCA boards per chassis with 64GB per DCA board) which is used for three basic purposes. It contains:
 - o Control Memory master copy on first two DCA boards (4-28GB each, mirrored)
 - o Data Cache Directories (500MB, one on each DCA board)
 - o Data Cache Regions (one per MPU board) allocated from all of the remaining space
- **VSD Local Memory (LM)** on each VSD board used for a process execution for each Intel processor core and a local copy of part of the system Control Memory.
- **FED and BED Local Memory** on each FED and BED board for temporarily buffering data blocks, as well as holding control information such as the LDEV to MPU mapping tables.

Cache System

The cache system is built on 2 to 16 DCA cache boards per VSP storage system (single or dual chassis). Each such DCA board has four banks of DDR3 DRAM, local battery power, and one or two SSDs for backup. These on-board SSD drives (31.5GB or 63GB each) are for use in backing up the entire memory space per board in the event of a storage system shutdown.

Each DCA board has four PCIe x4 links to the GSW boards. The aggregate raw GSW bandwidth per DCA board is 4GB/s send and 4GB/s receive. As such, each chassis, when populated with 8 DCA boards, provides a raw sequential memory bandwidth of 16GB/s send and 16GB/s receive. A two chassis system will have double that cache bandwidth, or 32GB/s send and 32GB/s receive.

VSD Board Local Memory

There is 4GB of local DDR2 DRAM on each VSD board. This local DRAM space is partitioned into five regions, with one region used by each of the VSD's Xeon's four cores for their private execution space plus a shared Control Memory region. The Control Memory region is a subset of the master Control Memory region in cache. It only contains the metadata and tables for those LDEVs that that VSD board's MPU manages. All writes to this Control Memory region are also immediately written to the master copy in cache. The majority of Control Memory accesses are expected to be reads. As such, much of what is needed is in the local high speed DRAM.

DXBF: FED and BED Board Local Memory

The local DRAM (also called the DXBF) found on each FED or BED board is used for buffering data, maintaining LDEV mapping tables (to the managing MPUs) and certain housekeeping and status tables. There are two GSW links per FED board and four per BED board over which pass job requests to MPUs and user data moving to or from cache.

FED Open Fibre Port Operations

I/O Request Limits and Queue Depths

Every Open Fibre path between the host and the storage system has a specific maximum instantaneous request capacity, known as the **Maximum I/O Request Limit**. [Note: The FICON environment is very different.] This is the limit to the *aggregate* number of requests (host tags) being directed against the individual LDEVs mapped to that host port. For the VSP, the average per port limit is about **2048 tags**. [This is handled differently for FICON, where the per-port limit is 480 for Open Exchange messaging.] However, when an Open Fibre FED port is placed into **external storage mode** (it becomes a SCSI *initiator*), the port I/O Request Limit drops to **384** tags in order to control the response times from the external storage system. The actual port limit is controlled by the individual MPUs. Each MPU actually has a tag limit of 256k entries for all of the LDEVs it manages from all of the host ports in use. So it is difficult to state what the actual limit could be as this is a dynamic value across all installed VSD boards. So we generally use 2048 tags per port (or 384 tags per external port) as a rule of thumb to work with.

LUN Queue Depth is server's maximum number of outstanding I/O requests (tags) **per LUN** per port. This is separate from the storage system's port **I/O Request Limit**. [Note that there is no concept of Queue Depth for FICON volumes.] A LUN queue depth is normally associated with random I/O workloads and high server thread counts since sequential workloads are usually operating with a very low thread count.

On the VSP, the per-LDEV rule-of-thumb queue depth is 32 per port but can be much higher (no real limit) in the absence of other workloads on that FED's ports for internal LDEVs. There isn't a queue depth limit per LDEV as such, but the usable limit will depend upon that Parity Group's ability to keep up with the workload demand.

In the case of external (Virtualized) LUNs, the VSP's internal LDEV queue depth (tags) can be set to 2-to-128 in the **Universal Volume Manager** GUI. Increasing the default queue depth value for external LUNs

from 8 up to 32 can often have a very positive effect (especially on response time) on OLTP-like workloads. The effective limit will depend on how many other concurrently active external LUNs there are on that port, as there is a fixed limit of 384 tags per external port.

External Storage Mode I/O (Open Fibre FED)

When using the Open Fibre FEDs, each port on the board may be configured as one of four modes: a **target** mode (accepting host requests), an **external** mode (driving virtualization requests), as **MCU Initiator** mode or as **RCU Target** mode.

The **external mode** is how other storage systems are attached to FED ports on a VSP. Front end ports from the external (secondary) storage system are attached to some number of VSP FED ports as if the VSP was a Windows server. In essence, those FED ports are operated as though they were a type of BED port. LUNs from attached storage systems that are visible on these external ports are then remapped by the VSP out through other specified FED ports that are attached to hosts (target mode). Each such external LUN will become an internal VSP LDEV that is managed by one of the MPUs.

As I/O requests arrive over host-attached ports on the VSP for an external LUN, the normal routing operations within the FED boards occur. The request is managed by the owner MPU in Control Memory tables, and the data blocks go into that MPU's data cache region. But the request is then routed to one or more other FED ports (initiator mode) that control the external paths where that LUN is located. The external storage system processes the request internally as though it were talking to a server instead of the VSP.

The two **MCU** and **RCU** port modes are for use with the Hitachi Universal Replicator and TrueCopy software that connects two Hitachi enterprise storage systems together for replication.

Cache Mode Settings

The external port (and all LUNs present on it) may be set to either "Cache Mode =ON" or "OFF" when it is configured in the VSP. This controls the write behavior. The "ON" setting has cache behavior identical to that of internal LDEVs, with immediate response to the server once the write data has been duplexed into the VSP's cache. The "OFF" setting directs the VSD to hold off on the write completion message to the server until the data is staged in cache in the *external* storage system. The "OFF" setting does not change any other cache behavior such as reads to the external LUNs. However, this change makes a significant difference when writing to slower external storage that presents a risk for high write pending cases to develop. "Cache Mode = ON" should normally not be used when high write pending rates are expected to occur in the external storage system. In general, the rule of thumb is to use "Cache Mode = OFF".

Other External Mode Effects

Recall from above that the Queue Depth for each external LUN can be modified to be between 2 and 128 in the Universal Volume Manager configuration screens. Increasing the default value from 8 up to 32 (probably the best choice overall), 64, or perhaps even 128 can often have a very positive effect (especially Response Time) on OLTP-like workloads. Along with this, recall that the maximum overall I/O Request Limit for an external port is 384 (not the host facing port average of 2048). If this "pass-through" rate is exceeded by the host, the requests will back up into the MPU queue for this LDEV until that MPU's overall tag limit of 256k is reached. At first the host will see rapidly rising response times per request until it hits a point (256k) where new host requests are held off.

The external port represents itself as a Windows server to the external storage system. Therefore, when configuring the *host type* on the ports on the virtualized storage system, it must be set to “Windows mode”.

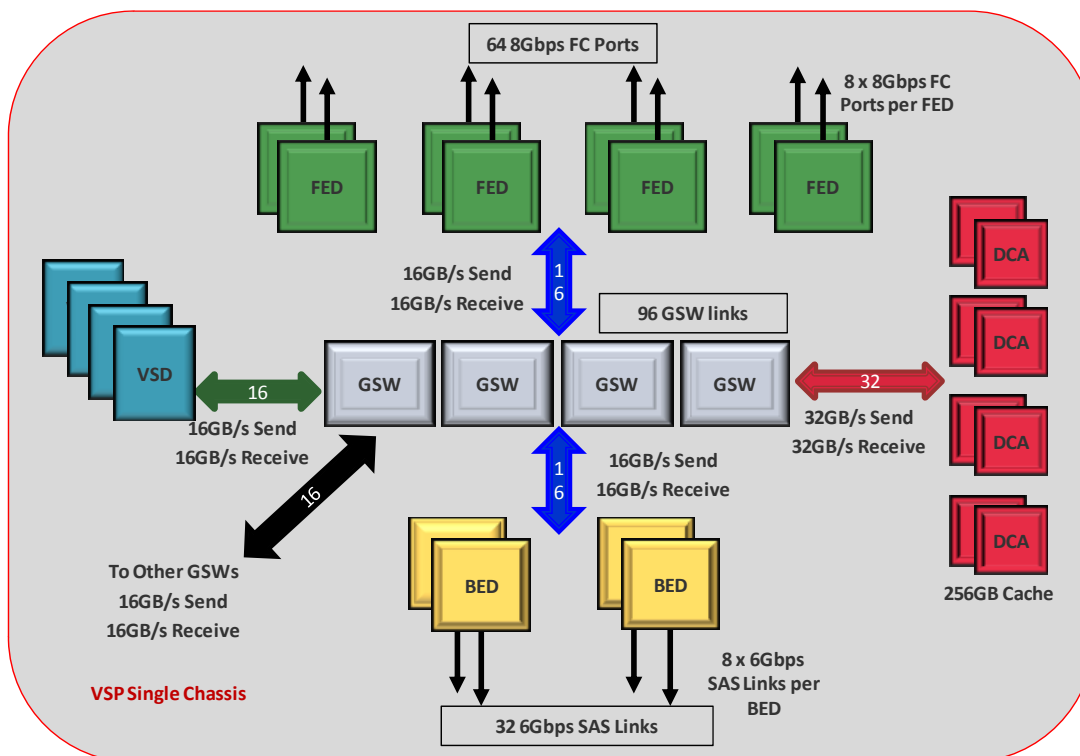
VSP Chassis Bandwidth Overview

The VSP is not being offered in specific models but as a basic model that is a starting point for configurations that are configured to meet a variety of customer requirements. The Figure below shows a single chassis storage system with all logic boards installed and with all cache DIMMs installed in each DCA board. The various peak wire speed bandwidths are shown for the different points in the storage system. Wire speed rates are what can be achieved by the bus protocol (PCIe in this case) for a short burst in a laboratory environment. But the protocol rates are what all vendors advertise since user achievable rates depend greatly on the environment and workloads. So the only useful reference points are usually the electrical limits of the various components.

VSP Bandwidths

The numbers inside the colored arrows (such as the red arrow to cache with a “32” indicate the number of Grid Switch links. The numbers next to them indicate the peak protocol for sending or receiving. Each such GSW path has a 1GB/sec send rate plus a concurrent 1GB/sec receive rate. So having 16 such links gives an aggregate of 16GB/sec send and 16GB/sec receive.

Figure 30. Overview of Single Chassis VSP Protocol Bandwidths



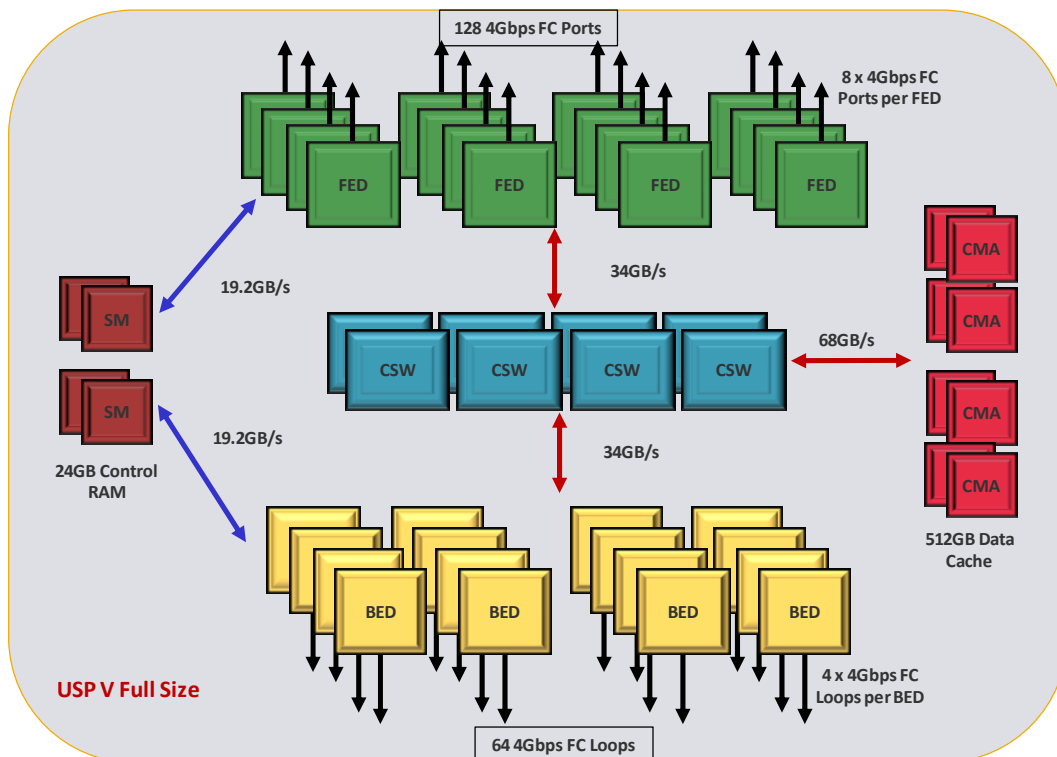
The peak overall rates (as send + receive GB/sec) for a fully loaded single chassis system under sustained heavy loads using all ports are:

- GSWs to FEDs: 16 + 16 GB/s
- GSWs to DCAs: 32 + 32 GB/s
- GSWs to VSDs: 16+16 GB/s
- GSWs to BEDs: 16 + 16 GB/s
- GSWs (Chassis-1) to GSWs (Chassis-2): 16 + 16 GB/s

USP V Bandwidths

In Figure 31 (showing a USP V), the numbers adjacent to the colored arrows indicate the *peak wire speed* rates for sending or receiving. On the CSW (Cache Switch) system of the USP V, a path was either in send mode or receive mode (half duplex). So a specified bandwidth is for the peak aggregate rates of these states. Each such CSW path has a 1064MB/sec transfer rate. So having 32 such ports gives an aggregate rate of 34GB/sec.

Figure 31. Overview of USP V's Protocol Bandwidths



The peak rates (shown as GB/sec) for a fully loaded USP V system under sustained heavy loads using all ports are:

- CSWs to FEDs: 34 GB/s
- CSWs to CMA (cache): 68 GB/s
- CSWs to BEDs: 34 GB/s
- FEDs to SM (Shared Memory): 19.2 GB/s
- BEDs to SM: 19.2 GB/s

General Storage Concepts

Understand Your Customer's Environment

Before recommending a storage design for customers, it is important to know how the product will address the customer's specific business needs. The factors that must be taken into consideration include: **Capacity, Performance, Reliability, Features and Cost** with respect to the storage infrastructure component. The more data you have, the easier it is to architect a solution as opposed to just selling another storage unit. The types of storage configured, including the drive types, the number of RAID Groups and their RAID levels, as well as the number and type of host paths is important to the solution.

RAID Levels and Write Penalties

The VSP storage system supports RAID levels 5, 6, and 10, where RAID-5 is the most space efficient.

Table 3.RAID Level Comparison

	RAID-10	RAID-5	RAID-6
Description	Data Striping and Mirroring	Data Striping with distributed parity	Data Striping with two distributed parities
Number of drives	4/8	4/8	8/16
Benefit	Highest performance with data redundancy; higher write IOPS per Parity Group than with similar RAID-5.	The best balance of cost, reliability, and performance.	Balance of cost, with extreme emphasis on reliability
Disadvantages	Higher cost per number of physical drives	Performance penalty for high percentage of Random Writes	Performance penalty for all writes

RAID-5 is a group of drives (RAID Group or Parity Group) with the space of one drive used for the rotating parity chunk per RAID stripe (row of chunks across the set of drives). If using a 7D+1P configuration (7 data drives, 1 parity drive), then you get 87.5% capacity utilization for user data blocks out of that RAID Group.

RAID-6 is RAID-5 with a second "parity" drive for a complex mathematical checksum. The second parity block includes all of the data chunks plus the first parity chunk for that row. This would be indicated as a 6D+2P construction (75% capacity utilization) if using 8 drives, or 14D+2P (87.5% utilization) if using 16 drives.

RAID-10 is a mirroring and striping mechanism. First, individual pairs of drives are placed into a mirror state. Then two of these pairs are used in a simple RAID-0 stripe. As there are four drives in the RAID Group, this would be represented as RAID-10 (2D+2D) and have 50% capacity utilization. RAID-10 (4D+4D) would be the same but uses eight drives. RAID-10 is not the same as RAID-0+1, although usage by many would lead one to think this is the case. RAID-0+1 is a RAID-0 stripe of N-disks mirrored to another RAID-0 stripe of N-disks. This would also be shown as 2D+2D for a 4-disk construction. However, if one drive fails, that RAID-0 stripe also fails, and the mirror then fails, leaving a user with a single

unprotected RAID-0 group. In the case of real RAID-10, one drive of each mirror pair would have to fail before getting into this same unprotected state.

The factors in determining which RAID level to use are cost, reliability, and performance. Table 3 shows the major benefits and disadvantage of each RAID type. Each type provides its own unique set of benefits so a clear understanding of your customer’s requirements is crucial in this decision.

Another characteristic of RAID is the idea of “write penalty”. Each type of RAID has a different back end physical drive I/O cost, determined by the mechanism of that RAID level. The table below illustrates the trade-offs between the various RAID levels for write operations. There are additional physical drive reads and writes for every application write due to the use of mirrors or XOR parity.

Note that SATA drives are usually deployed with RAID-6 to protect against a second drive failure within the Parity Group during the lengthy drive rebuild of a failed drive. Also note that you must deploy far more SATA drives than SAS drives to be able to meet the same level of IOPS performance.

Table 4. RAID Write Penalties

SAS / SSD/ FMD	HDD IOPS per Host Read	HDD IOPS per Host Write
RAID-10	1	2
RAID-5	1	4
RAID-6	1	6
SATA		
RAID-10	1	4
RAID-5	1	6
RAID-6	1	9

To protect data, in the case of SATA drives, there are additional physical I/O operations per write in order to compare what was just written to drive with the data and parity blocks held in cache. With RAID-6, there are three such blocks: data, parity1, and parity2.

Table 5. Break Down of RAID Level Write Costs (Fibre Channel drives)

	IOPS Consumed per host I/O request
RAID-10	1 Data Write, 1 mirrored Data Write
RAID-5	2 reads (1 data, 1 parity), 2 writes (1 data, 1 parity)
RAID-6	3 reads (1 data, 2 parity), 3 writes (1 data, 2 parity)
	SATA writes: also read and compare data and parity writes (mirror)

Parity Groups and Array Groups

On Hitachi enterprise products the terms **Parity Group** and **Array Group** are used when configuring storage. The Array Group is the set of four associated drives, while the Parity Group is the RAID level applied to either 1 or 2 sets of drives. However, in general use in the field, the terms Array Group and Parity Group are often used interchangeably.

RAID Chunks and Stripes

The Parity Group is a logical mechanism that has two basic elements: a virtual block size from each drive (a **chunk**) and a row of chunks across the group (the RAID **stripe**). The chunk size is 256KB on the VSP, but two such chunks per drive are used at a time so the effective chunk size is really **512KB**. The stripe size is the sum of the chunk sizes across a RAID Group. This only counts the “data” chunks and not any mirror or parity space. Therefore, on a RAID-6 group created as 6D+2P (eight drives), the stripe size would be 3072MB (512KB x 6).

Note that some industry usage replaces *chunk* with “stripe size”, “stripe depth”, or “interleave factor”, and *stripe size* with “stripe width”, “row width” or “row size”.

Note that on all current RAID systems, the chunk is a primarily unit of protection and layout management: either the parity or mirror mechanism. I/O is not performed on a chunk basis as is commonly thought. On Open Systems, the entire space presented by a LUN is a contiguous span of 512 byte blocks, known as the Logical Block Address range (LBA). The host application makes I/O requests using some native request size (such as a file system block size), and this is passed down to the storage as a unique I/O request. The request has the starting address (of a 512 byte block) and a length (such as the file system 8KB block size). The storage system will locate that address within that LUN to a particular drive address, and read or write only that amount of data – not that entire chunk. Also note that this request could require two drives to satisfy if 2KB of the block lies on one chunk and 6KB on the next one in the stripe.

Because of the variations of file system formatting and such, there is no way to determine where a particular block may lie on the raw space presented by a volume. A file system will create a variety of metadata in a quantity and distribution pattern that is related to the size of that volume. Most file systems also typically scatter writes around within the LBA range – an outdated hold-over from long ago when file systems wanted to avoid a common problem of the appearance of bad sectors or tracks on drives. What this means is that attempts to align application block sizes with RAID chunk sizes is a pointless exercise.

The one alignment issue that should be noted is in the case of host-based Logical Volume Managers. These also have a native “stripe size” that is selectable when creating a logical volume from several physical storage LUNs. In this case, the LVM stripe size should be a multiple of the RAID chunk size due to various interactions between the LVM and the LUNs. One such example is the case of large block sequential I/O. If the LVM stripe size is equal to the RAID chunk size, then a series of requests will be issued to different LUNs for that same I/O, making the request appear to be several random I/O operations to the storage system. This can defeat the storage system’s sequential detect mechanisms, and turn off sequential prefetch, slowing down these types of operations.

LUNS (host volumes)

On a VSP, when space is carved out of a Parity Group and made into a volume, it is then known as a Logical Device (LDEV). Once the LDEV is mapped to a host port for use by a server, it is known as a LUN (Logical Unit Number) and is assigned a certain *World Wide Name* if using fibre channel interfaces on the storage system. On an iSCSI configuration, the LUN gets a name that is associated with an NFS mount point. Note that general usage has turned the term of “LU” into “LUN”.

Number of LUNs per Parity Group

When configuring a VSP, one or more LDEVs may be created per Parity Group, but the goal should be to clearly understand what percentage of that group's overall capacity will contain active data. In the case where multiple hosts attempt to simultaneously use LUNs that share the same physical drives in an attempt to fully utilize capacity, seek and rotational latency may be a performance limiting factor. In attempting to maximize utilization, RAID groups should contain both active and less frequently used LUNs. This is true of all physical drives regardless of size, RAID level, and physical characteristics. Note that individual LDEVs from the same Parity Group will be typically (automatically) managed by different MPUs.

Port I/O Request Limits, LUN Queue Depths, and Transfer sizes

There are three aspects of I/O request handling per storage path that need to be understood. At the port level, there are the mechanisms of an **I/O request limit** and a **maximum transfer size**. At the LUN level, there is the mechanism of the **maximum queue depth**.

Port I/O Request Limits

Each server and storage system has a particular **port I/O request limit**, this being the number of total outstanding I/O requests that may be issued to a storage system **port** (not a LUN) at any point in time. This limit will vary according to the server's Fibre Channel HBA and its driver, as well as the limit supported by the target storage port. As switches serve to aggregate many HBAs to the same storage system port, the operating limit will be the lower of these two points. In the case of a SAN switch being used to attach multiple server ports to a single storage port (fan-in), the limit will most likely be the limit supported by the storage port.

On the VSP, the rule of thumb average I/O request limit per FED port is 2048. This means that, at any one time, there can be up to 2048 total active host I/O commands (tags) queued up for the various LUNs visible on that FED port. The actual limit is determined by how many requests (tags) are currently in the queue on all of the VSD boards for the installed host ports. Each board has a limit of 256k tags to use in managing all FED and BED I/O requests.

Most environments don't require that all LUNs be active at the same time. As I/O requests are application driven, this must be taken into consideration. Understanding the customer's requirements for performance will dictate how many LUNs should be assigned to a single port. Environments that have low I/O demands overall will allow a large number of LUNs to be assigned per port without routinely exceeding the queue limit for that port. On the other hand, environments with highly active devices and multiple I/O request should be designed with fewer LUNs per port, where additional ports will be required in order to satisfy the I/O demand.

Port I/O Request Maximum Transfer Size

There is also a host tunable parameter at the driver level that controls the **maximum transfer size** of a single I/O operation. This is usually a per-port setting, but it also might be a per-host setting. The defaults for this can range from fairly small (like 32KB) to something midsize (128KB). The maximum is usually 1MB-16MB. It is probably a best practice to set this to the largest setting that is supported on the host. The maximum transfer size that a VSP will accept on an Open System FED port is 16MB. This value controls how application I/O requests get processed. If one used a large application block size (say 256KB) and the port/LUN default was just 32KB, then each request would be broken down (fragmented)

into 8 * 32KB requests. This creates additional overhead in managing the application request. The use of a large maximum transfer size such as 1024KB will often be readily apparent on the performance of the system.

LUN Queue Depth

There is a host port limit known as the **LUN maximum queue depth**. This is the limit of how many outstanding I/O requests may be scheduled by a host for each LUN on an HBA port. On a host HBA, the device queue depth limit is usually configurable as a driver tunable, usually on a per-port basis rather than a global setting. On Hitachi storage systems, there is no actual limit per LUN. The usable queue depth per LUN when there are multiple simultaneously active LUNs from the same RAID group limit will be determined by physical I/O capacity of that group of drives.

Mixing Data on the Physical drives

Physical placement of data by Parity Groups (not just the LDEVs from the same Parity Group) is extremely important when the data access patterns differ. Mixing highly “write intensive” data with high “read intensive” data will cause both to suffer performance degradation. This performance degradation will be much greater when using RAID-5 or RAID-6 due to the increase in back end drive operations required for writes. When using SAS drives, there are two physical I/Os required for each random write for RAID-10, four for RAID-5, and six for RAID-6. When using SATA drives and RAID-6, there are three additional physical I/Os required due to the read verification after writing (data and parity/parity).

Workload Characteristics

Most applications, from a drive storage system point of view, can be categorized into one of seven types of application profiles (though there are the extreme cases that are not covered here):

- Online Transactional Processing – Random Read/Write to data files, with Sequential Read/Write to Log functions
- Decision Support Systems – Sequential or Random Read, depending on data access method, post ETL
- General Purpose File systems – Random Read/Write with potentially some Sequential I/O
- Messaging systems – Random Read/Write
- Web Server systems – Mixed Random and Sequential with various block sizes.
- Content Rich Media Streaming and/or Video Streaming – Sequential Read/Write
- High-performance Computing – Sequential Read/Write

The applications that do not directly fit into one of these seven categories will most likely be a combination of two or more of these categories.

Selecting the Proper drive Form Factor

In all cases, distributing a workload across a higher number of small-capacity, high-RPM drives will provide better performance in terms of full random access. Even better results can be achieved by distributing the workload over a higher number of small LUNs where the LUNs are the only active LUNs

in the Parity Group. When cached data locality is low, multiple small-capacity high-RPM drives should be used.

One must also take into consideration the case where a system that is only partially populated with 15k RPM drives will be able to provide a much higher aggregate level of host IOPS if the same budget is applied to lower cost 10k RPM drives. For instance, if there is a 100% increase in the cost of the 15k drives, then one could install twice as many drives of the 10k variety. The individual I/O will see some increase in response time when using 10k drives, but the total amount of IOPS available in the system will be much higher.

Mixing I/O Profiles on the Physical drives

Mixing large-block sequential I/O with small-block random I/O on the same physical drives can result in poor performance. This applies to both read and write I/O patterns. This problem can occur at the LUN level or RAID Group level. In the first case, with a single large LUN, files with different I/O profiles will result in poor performance due to lack of sequential detect for the large block sequential I/O requests. In the second case, where multiple LUNs share a Parity Group, files having different I/O profiles will result in sequential I/O dominating the drive access time, due to pre-fetching, thereby creating high response times and low IOPS for the small-block random I/O. The resolution to these two issues is to use different LUNs and different Parity Groups for different data types and I/O profiles.

Front end Port Performance and Usage Considerations

The flexibility of the front end ports is such that several types of connections are possible. A couple of port usage points are considered:

- Port fan-in and fan-out
- I/O profile mixes on a port

Host Fan-in and Fan-out

With a SAN, the ability to share a port or ports among several host nodes is possible, as is configuring a single host node to multiple storage ports. Fan-in is one of the great promises of fiber channel based SAN technologies—the sharing of costly storage resources. Several host nodes are fanned *into* a single storage port, or a host node has fanned out to several storage ports.

Fan-in

Host Fan-in refers to the consolidation of many host nodes into one or just a few storage ports (many-to-one). Fan-in has the potential for performance issues by creating a bottleneck at the front end storage port. Having multiple hosts connected to the same storage port does work for environments that have minimal performance requirements. In designing this type of solution, it is important to understand the performance requirements of each host. If each host has either a high IOPS or throughput requirement, it is highly probable that a single 8Gbps FC-AL port will not satisfy their aggregate performance requirements.

Fan-out

Fan out allows a host node to take advantage of several storage ports (and possibly additional port processors) from a single host port (one-to-many). Fan-out has a potential performance benefit for small block random I/O workloads. This allows multiple storage ports (and their queues) to service a smaller number of host ports. Fan-out typically does not benefit environments with high throughput (MB/sec) requirements due to the transfer limits of the host bus adapters (HBAs).

Mixing I/O Profiles on a Port

Mixing large-block sequential I/O with small-block random I/O can result in poor performance. This applies to both read and write I/O patterns. This problem can occur at the LUN level or Parity Group level. In the first case, with a single large LUN, files with different I/O profiles will result in poor performance due to lack of sequential detect for the large block sequential I/O requests. In the second case, where multiple LUNs share a RAID Group, files having different I/O profiles will result in sequential I/O dominating the drive access time, due to pre-fetching, thereby creating high response times and low IOPS for the small-block random I/O. The resolution to these two issues is to use different LUNs and different storage system Groups for different data types and I/O profiles.

Summary

The VSP offers great flexibility and it should perform extremely well in any environment. At the hardware level, it offers considerably more IOPS power than any other enterprise product on the market. The Universal Storage Platform V already delivered best-of-class sustained Sequential performance (about three or four times higher than any of the competition with >11GB/s Read, and >6GB/s write), and the VSP also delivers this level of sequential performance.

The VSP system is far more flexible than the previous USP V design. Some of the issues that required careful configuration that no longer matter are:

- Since most of the FED port I/O processing occurs within the VSD board, there is no need to use the previous “port skipping” technique within a FED board to avoid overload of the local processors (MPs on the USP V). One will still need to use “FED skipping” to even out the loads per FED board.
- Since the VSD boards control most of the I/O process, there is no need to manage port assignments on a FED by type (host target, external initiator, HUR Initiator, HUR Target) to insure that most or all ports (on the same board) are operated in the same **mode** in order to achieve efficient “MP Workload Sharing” within a FED board. All ports on a USP V FED board do need to operate in the same mode in order to help each other.
- On a USP V, IOPS scaling required addition of more FED boards or BED boards beyond what was actually needed in order to make use of the port skipping technique of reducing the loads on the FED MPs. On the VSP, the IOPS power is scaled by the addition of VSD boards. Everything else can normally be “right sized” to what a customer needs.
- On the USP V, the write pending rate triggers were global, unless using the special 454 mode that would average the rates among the CLPR cache partitions (but you must be using CLPRs to do so). On the VSP, the write pending triggers are per MPU per CLPR. Other MPUs are not

affected by an MPU with high write pending rates, although the shared physical resources (such as cache paths) will be affected.

- The VSP provides far more IOPS power in a smaller footprint than a USP V, with greatly reduced power consumption.

It is important for users to understand the two configuration methods available with the *Hitachi Dynamic Provisioning* feature and the new *Hitachi Dynamic Tiering* feature within the HDP mechanism. The Hitachi Dynamic Provisioning feature can be used for configuring Thin Provisioning and for the creation of High Performance volumes. While only certain applications will benefit from the Thin Provisioning aspect, most can take advantage of the high performance per volume feature. The use of HDT can offer great flexibility in managing performance tiers within a storage system.

It is vital, however, that Hitachi Data Systems' sales personnel, technical support staff, value-added resellers, and others who are responsible delivery of solutions, invest the time required to design the best possible solution to meet each customer's unique requirements, whether it be capacity, reliability, performance or cost. To assist in delivering the highest quality solution, Hitachi Data Systems' Global Solution Services should be engaged.

Appendix 1. Open Systems RAID Mechanisms

This section describes how the Open-V Open Systems Groups function on both the Universal Storage Platform V and the VSP. The track size (chunk) used for Open-V is 256KB. The available 9900 Lightning style Open-X emulation is available, with its track (chunk) size of 48KB. One should avoid mixing Open-X and the standard Open-V emulations on the same storage system.

When using generic RAID-5, one would normally calculate the stripe width (row size) by multiplying the RAID chunk size (or “stripe size”) by 7 (if 7D+1P). In the case of Open Systems (Open-V volumes), the VSP uses a native chunk size of 256KB. Hence you would expect a row size of 1,792KB. However, the VSP manages RAID-5 somewhat differently than any other vendor. For both VSP RAID-5+ types, there are two adjacent 256KB RAID chunks taken from the *same* drive before switching to the next drive in the row, so the effective chunk size is 512KB. This is illustrated in the two Figures below. Thus, there are twice as many chunks per row as normal. So the VSP’s RAID-5+ 7D+1P stripe width is actually 3,584KB (7 x 512KB). The VSP’s RAID-5+ (3D+1P) stripe width is actually 1,536KB (3 x 512KB).

Figure 32. RAID-5+ (7D+1P) Layout (3,584KB stripe width).

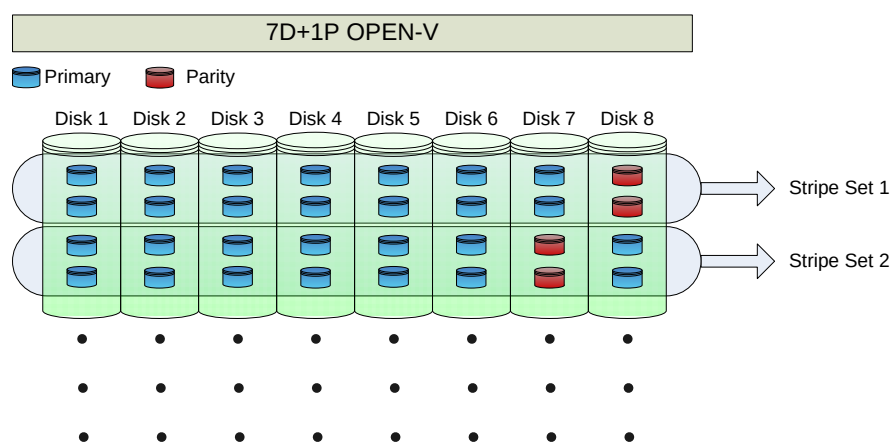
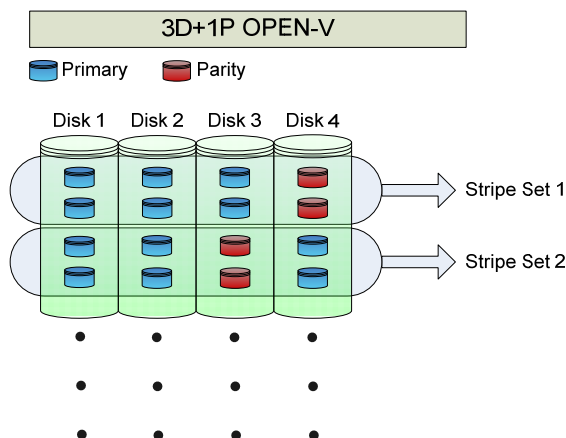


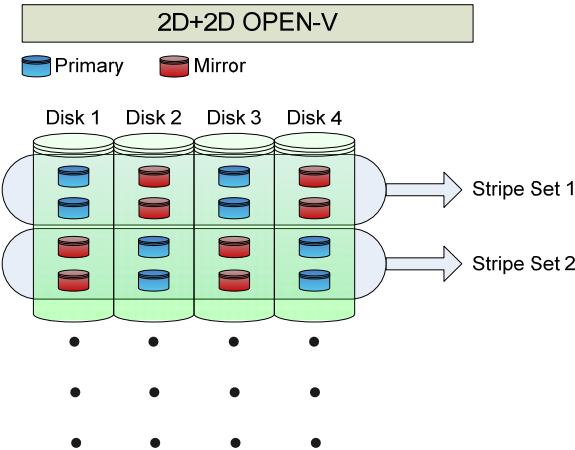
Figure 33. RAID-5+ (3D+1P) Layout (1,536KB stripe width).



When using generic RAID-10, one would normally calculate the stripe width (row size) by multiplying the RAID chunk size (or “stripe size”) by 2 (if 2D+2D). In the case of Open Systems (Open-V volumes), the VSP uses a native chunk size of 256KB. Hence you would expect a row size of 512KB.

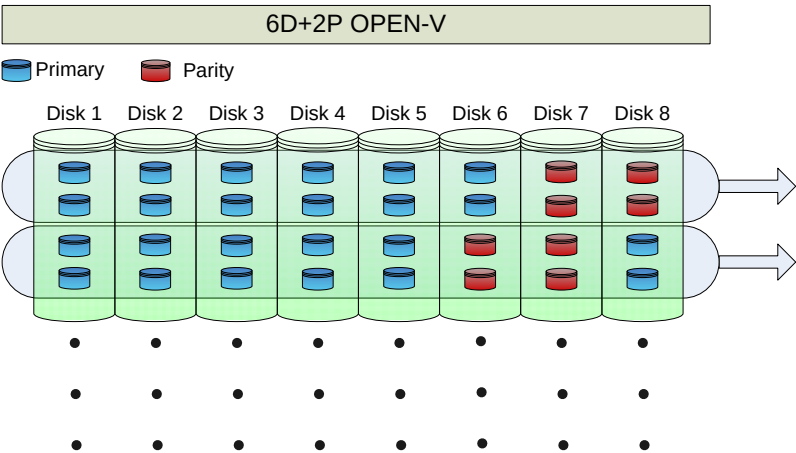
However, the VSP manages RAID-10+ in the same manner as RAID-5+, where there are two adjacent RAID chunks taken from the *same* drive before switching to the next drive in the row. This is illustrated in the next Figure. Thus, there are twice as many chunks per row as normal. So the VSP’s RAID-10+ (2D+2D) is 1,024KB (4 x 512KB).

Figure 34. RAID-10+ (2D+2D) Layout (stripe width 1024KB).



The VSP manages RAID-6+ in the same manner as RAID-5+, where there are two adjacent RAID chunks taken from the *same* drive before switching to the next drive in the row. This is illustrated in the Figure below. Thus, there are twice as many chunks per row as normal. So the VSP’s RAID-6 6D+2P stripe width is 3,072KB (6 x 512KB). The larger RAID-6 (14D+2P) would look similar but have 16 disks, and the stripe size would be 7,168MB (14 x 512KB).

Figure 35. RAID-6+ (6D+2P) Layout (stripe width 3,072KB)



Appendix 2. Mainframe 3390x and Open-x RAID Mechanisms

This section describes how the 3390x (mainframe) and Open-X (9900V Lightning style Open Systems volumes) Parity Groups function on both the Universal Storage Platform V and the VSP. The track size (chunk) used for 3390x is 58KB, and the Open-X track (chunk) size is 48KB. One should avoid mixing Open-X and the standard Open-V on the same storage system.

RAID-5+

When using generic RAID-5, one would normally calculate the stripe width (row size) by multiplying the RAID chunk size (or “stripe size”) by 7 (if 7D+1P). In the case of mainframe emulation (3390-x volumes), the VSP uses a native track size of 58KB. Hence you would expect a row size of 406KB (7 x 58KB). For the Open-X emulation, the track size is 48KB, with an expected row size of 336KB (7 x 48KB).

However, the VSP manages RAID-5 somewhat differently than any other vendor. For both VSP RAID-5+ types, there are eight adjacent tracks taken from the *same* drive before switching to the next drive in the row. This is illustrated in the two Figures below. The VSP’s RAID-5+ 7D+1P stripe width for 3390-X is 3,248KB (7 x 8 x 58KB), and for Open-X it is 2,688KB (7 x 8 x 48KB). The VSP’s RAID-5+ (3D+1P) stripe width for 3390-X is 1,392KB (3 x 8 x 256KB), and for Open-X it is 1,152KB (3 x 8 x 48KB).

Figure 36. RAID-5+ (7D+1P) Layout, 3390-X and Open-X

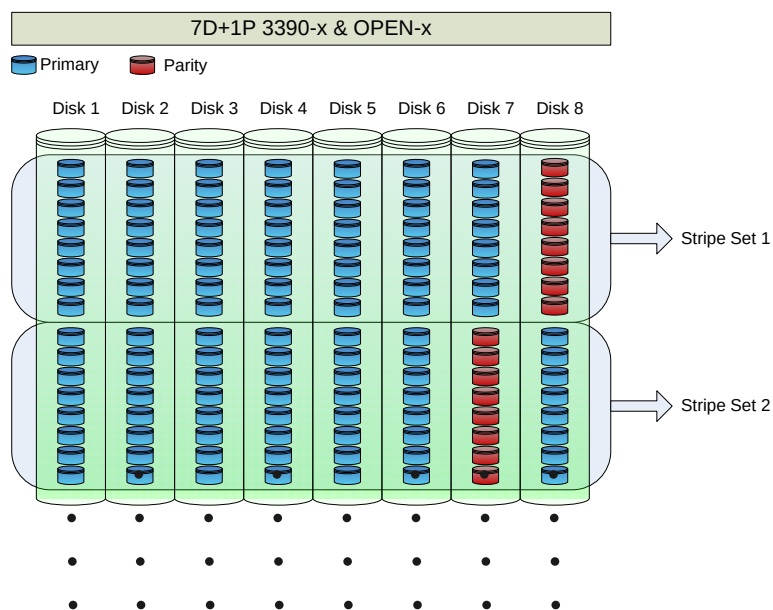
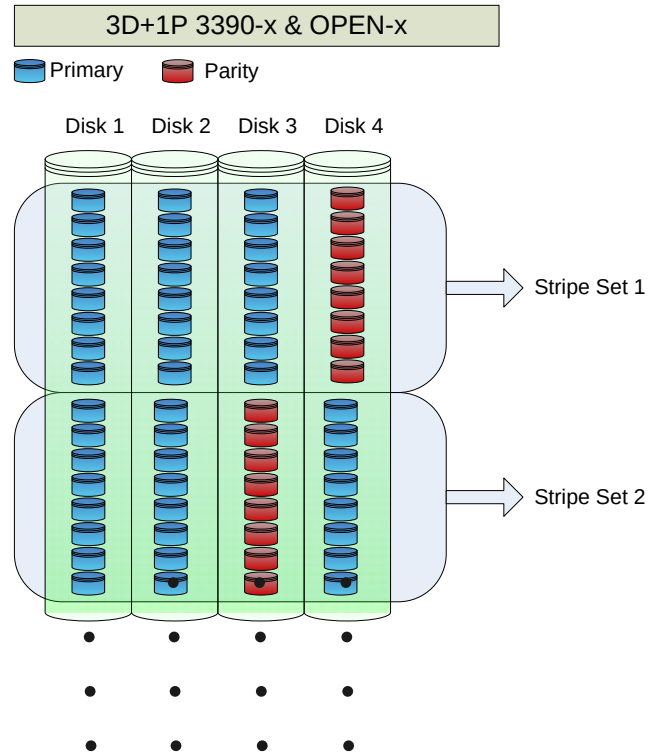


Figure 37. RAID-5+ (3D+1P) Layout, 3390-X and Open-X

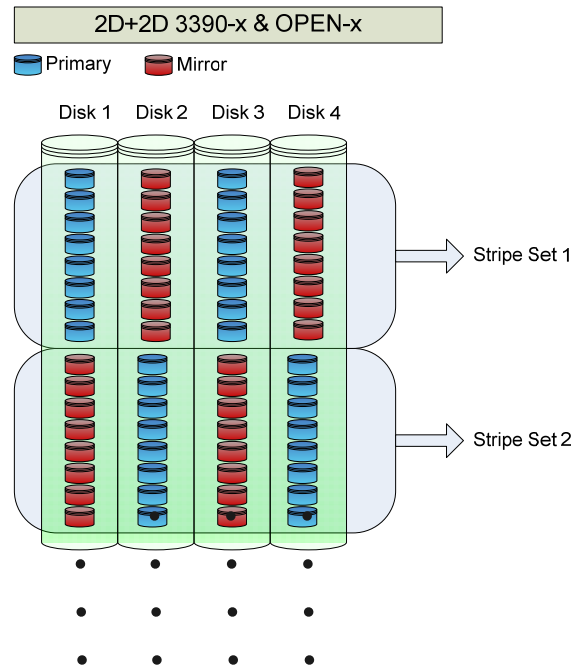


RAID-10+

When using generic RAID-10, one would normally calculate the stripe width (row size) by multiplying the RAID chunk size (or “track size”) by 2 (if 2D+2D). In the case of mainframe emulation (3390-x volumes), the VSP uses a native track size of 58KB. Hence you would expect a row size of 116KB (2 x 58KB). For the Open-X emulation, the track size is 48KB, with an expected row size of 96KB (2 x 48KB).

However, the VSP manages RAID-10+ in the same manner as RAID-5+, where there are eight adjacent tracks taken from the *same* drive before switching to the next drive in the row. This is illustrated in the two Figures below. So the VSP’s RAID-10+ (2D+2D) is 928KB (2 x 8 x 58KB). The VSP’s RAID-10+ stripe width for Open-X is 768KB (2 x 8 x 48KB).

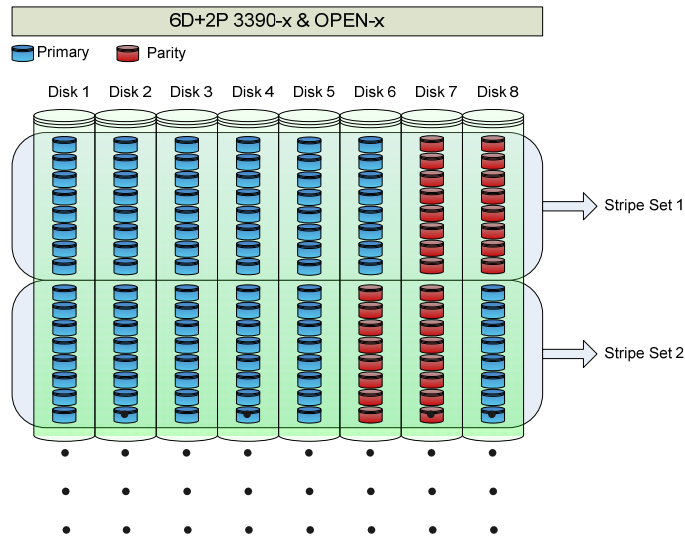
Figure 38. RAID-10+ (2D+2D) Layout, 3390-X and Open-X.



RAID-6+

The VSP manages RAID-6+ in the same manner as RAID-5+ where there are eight adjacent tracks taken from the *same* drive before switching to the next drive in the row. This is illustrated in the Figure below. Thus, there are twice as many chunks per row as normal. So, for 3390-X, the VSP’s RAID-6 6D+2P stripe width is actually 2,784KB (6 x 8 x 58KB), and for Open-X it is 2,304KB (6 x 8 x 48KB).

Figure 39. RAID-6+ (6D+2P) Layout, 3390-X and Open-X.



Appendix 3. Putting SATA drive Performance into Perspective

By Ian Vogelesang (A generic discussion of SATA)

The fact that we already worry about “hot spots” (busy parity groups) means that Fibre Channel drives are already quite busy. In other words, we generally adjust workloads on Fibre Channel drives to stay within their IOPS handling capability.

For example, the performance benefit of Dynamic Provisioning is that the I/O activity associated with a large number of host DPVOLs can be combined and distributed over many parity groups. This evens out the activity over those many parity groups and thus eliminates hot spots.

Thus, by eliminating hot spots, Dynamic Provisioning lets you raise average parity group utilization (% busy) somewhat. But what does “raise parity group utilization” mean? It means “put more data on each drive”.

If we combine the data from two separate hosts or two applications together and store them on one parity group, we are also combining the associated activity (IOPS) together.

Thus if we put twice as much data *of the same type* on one drive, we will also double the activity (IOPS) on the drive. Notice that there is the concept of the type of data in terms of its intensity of access. Some types of data are infrequently accessed, and others are heavily accessed. This is a fundamental characteristic of the data, no matter what type of drive the data is used to store the data. The activity is associated with the data, and if you have twice the data, you will have twice the associated activity.

We call this intrinsic measure of the activity associated with a given amount of data the **Access Density**, and it is measured in IOPS per GB. For example, if you have 100 IOPS associated with 200GB of data the access density for that data is 100 IOPS per 200GB of data, or 0.5 IOPS per GB.

Compared to Fibre Channel drives:

- SATA drives have a much lower IOPS capability on a per drive basis.
- SATA drives have a much higher storage capacity on a per drive basis.

Thus to figure out if it's cost-effective to use SATA drives, you need to figure out not only how many SATA drives it will take to hold the data, but also how many SATA drives it will take to handle the IOPS that come with the data.

In general we can't put most “normal” types of data on SATA drives, because SATA drives have a *much* lower IOPS limit than that of Fibre Channel drives.

Let's look at some numbers:

Each I/O operation begins with mechanical positioning, which starts with a seek operation (to move the access arm into position) followed by latency (waiting for the drive to turn until the starting point of the target data comes under the head).

The average seek time for a drive is generally longer than the average latency (1/2 turn), so in the time it takes to do the mechanical positioning, the drive makes more than one full rotation. However, a

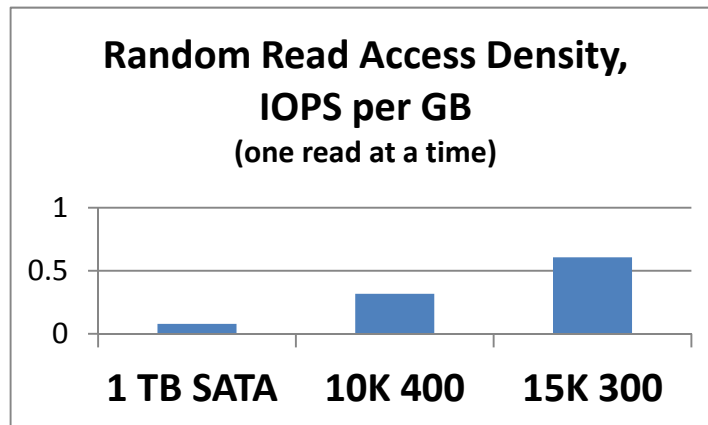
random I/O typically processes a 4 KB or 8 KB block, which is on the order of one percent of the storage capacity of one drive track. The capacity of one track varies a lot between drive types and the inner and outer zones of the drive, but in general the transfer time for a random I/O block is on the order of less than 1% of the total I/O service time, so random I/O is 99% mechanical positioning.

Nominal “one at a time” IOPS capability of sample drive types:

	1TB SATA drive	400GB 10,000 RPM Fibre Channel drive	300GB 15,000 RPM Fibre Channel drive
Average read seek time	8.5 ms	4.9 ms	3.5 ms
Average latency This is ½ of a rotation.	4.2 ms	3.0 ms	2.0 ms
Total mechanical positioning time (read)	12.7 ms	7.9 ms	5.5 ms
Read IOPS capability This is the number of (read) mechanical positioning operations (IOPS) that can be done in one second, “one at a time”.	79 IOPS	127 IOPS	182 IOPS

When we factor in the amount of data that is stored on the drive, we can calculate the base “one at a time” read access density capability of each type of drive:

	1TB SATA drive	400GB 10,000 RPM Fibre Channel drive	300GB 15,000 RPM Fibre Channel drive
Read IOPS capability Number of (read) mechanical positioning operations (IOPS) that can be done in one second, “one at a time”.	79 IOPS	127 IOPS	182 IOPS
Storage capacity, GB	1,000GB	400GB	300GB
Nominal read access density capability	.08 IOPS per GB	.32 IOPS per GB	.61 IOPS per GB



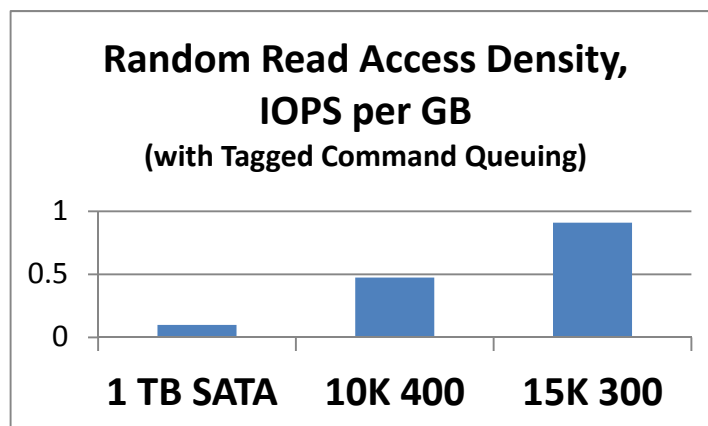
What we can see here is that if we fill each drive with data, 10 K RPM drives are capable of handling data that is 4 times as active as the data we can put on a SATA drive, and 15K RPM drives are capable of handling data that is 8 times as active.

It gets worse.

I/O operations are not issued one-at-a-time. Multiple I/O operations (each identified by a “tag”) are issued at once to each drive, in order to let the drive decide what order to perform the I/O operations in, so as to minimize the amount of seeking and rotational delay necessary. This capability to reduce average mechanical positioning time is called **Tagged Command Queuing**, or **TCQ**.

Fibre Channel drives have substantially more microprocessor horsepower than SATA drives, and as a result, they get substantially higher boosts in throughput through the use of TCQ. The amount of throughput boost from TCQ varies by workload, but for the purposes of illustration, it’s reasonable to show at 25% boost in IOPS capability using TCQ for SATA, and a 50% boost for Fibre Channel.

Now a more realistic comparison for read Access Density capability is shown:

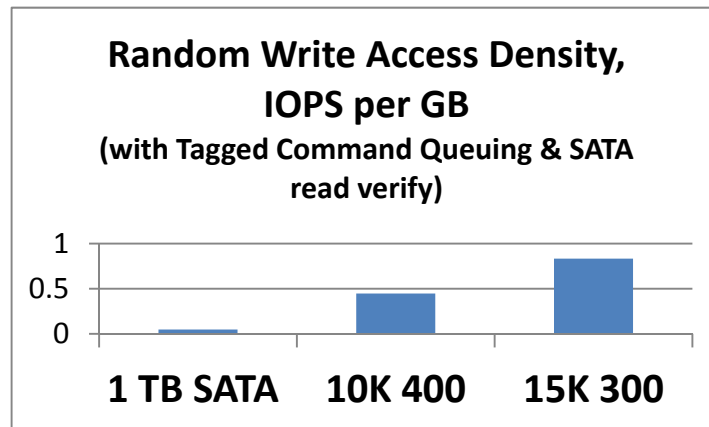


Now the Access Density capability of 10 K RPM drives is roughly 5x that of SATA, and 15K RPM drives roughly 10x that of SATA.

It gets even worse.

The use of SATA drives in storage system applications is relatively recent, and protecting the integrity of customer data is paramount. Therefore, Hitachi storage systems perform a read-verify operation after

every write with SATA drives. This adds a “random read” positioning time after every write operation. Note that this is not “one extra rotation”, but truly a full random read positioning, as in general multiple I/O operations are queued in the drive at once, and therefore the read-verify operation goes in the queue after the write completes, and other I/O operations in general will intervene between the write and subsequent read verify.



For random writes, the 10K RPM drive is capable of handling about 10x the access density of the SATA drive, and the 15K RPM drive about 18 times that of the SATA drive.

These numbers are only rough estimates, but what you can clearly see here is that in order to realize the cost effectiveness of SATA drives, that is, in order to take advantage of the storage capacity of SATA drives they need to be filled with data. The only type of data that can fill a SATA drive is data that is approximately an order of magnitude (10 times) less active than data that would normally be put on Fibre Channel drives.

Failure to understand this 10 times difference in access density capability can result in serious SATA drive deployment problems. If normal Fibre Channel drive-type data is placed on SATA drives, the result will usually be that the SATA drives cannot keep up with the IOPS demand, and storage system cache will fill up with data waiting to be written to the SATA drives. In order to treat the symptom (excessive “write pending”), SATA drive workloads can be placed in their own Cache Logical Partition (CLPR). Even though this may isolate other workloads from the problem, the SATA drives will still be a bottleneck constraining the throughput of the application that is using them.

The summary on SATA drives is that if they are used for traditional application data, there are only two scenarios:

- 1) Cost was reduced by putting over 100GB of normal data on a 1 TB SATA drive. Excessive write pending forces the SATA drives to be isolated in a CLPR. Cost was reduced, but throughput is drastically reduced.
- 2) Less than 100GB of data was placed on each SATA drive. The extra money (over and above what it would have cost for Fibre Channel drivers) was spent in order to provide enough SATA drives to handle the IOPS. For this price premium the result is slower response time and higher electrical power costs.

Thus only if you have a type of workload that has at most 1/10th the access density of normal workloads will SATA drives be cost effective.

Appendix 4. LDEVs, LUNs, VDEVs and More

Logical Devices (VSP, USPV)

This section provides a general overview of logical and physical devices on the past few generations of Hitachi enterprise systems, much of this going back before the Universal Storage Platform systems. This section was developed for the USP V generation, but still applies with few changes to the VSP.

An LDEV is a logical volume internal to the storage system, which can hold host data. LDEVs are “carved” from a logical container called a VDEV. In the diagram below, an internal VDEV is shown, which maps to the formatted space (net of parity) in a Parity Group.

Figure 1. Overview of Logical Devices.

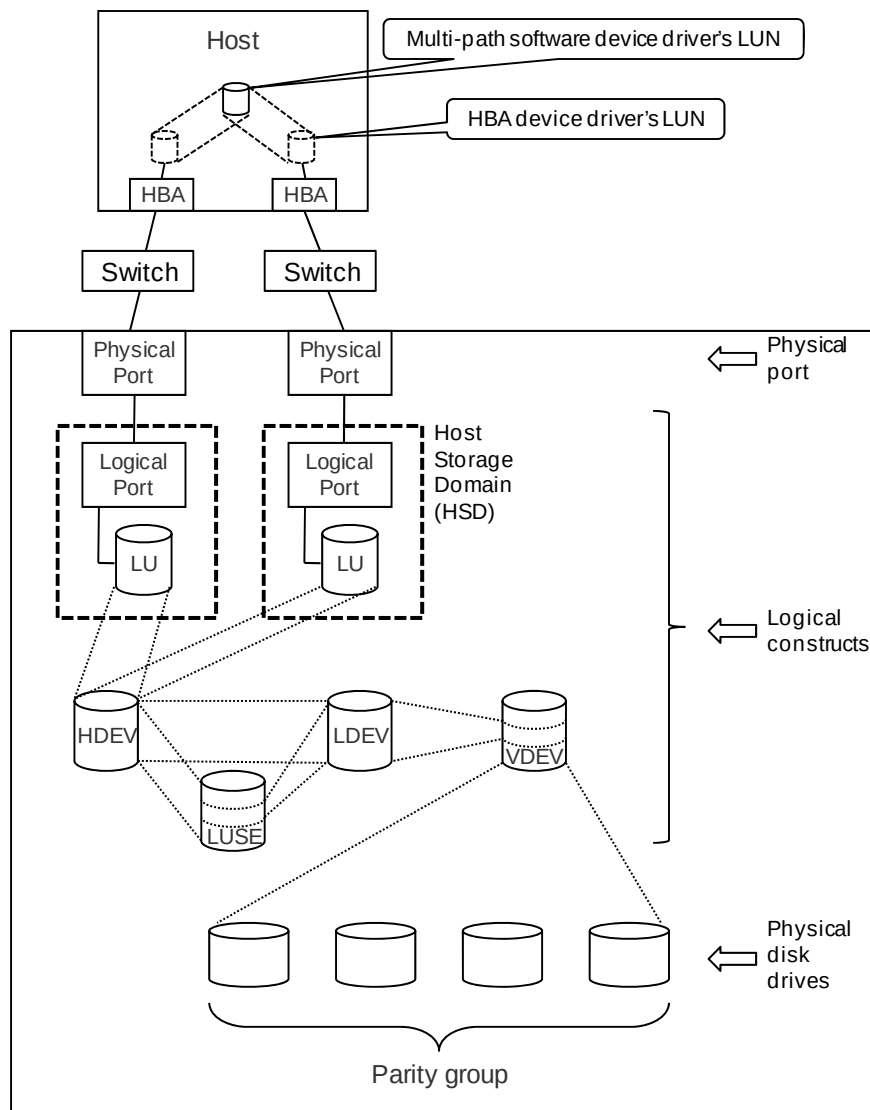
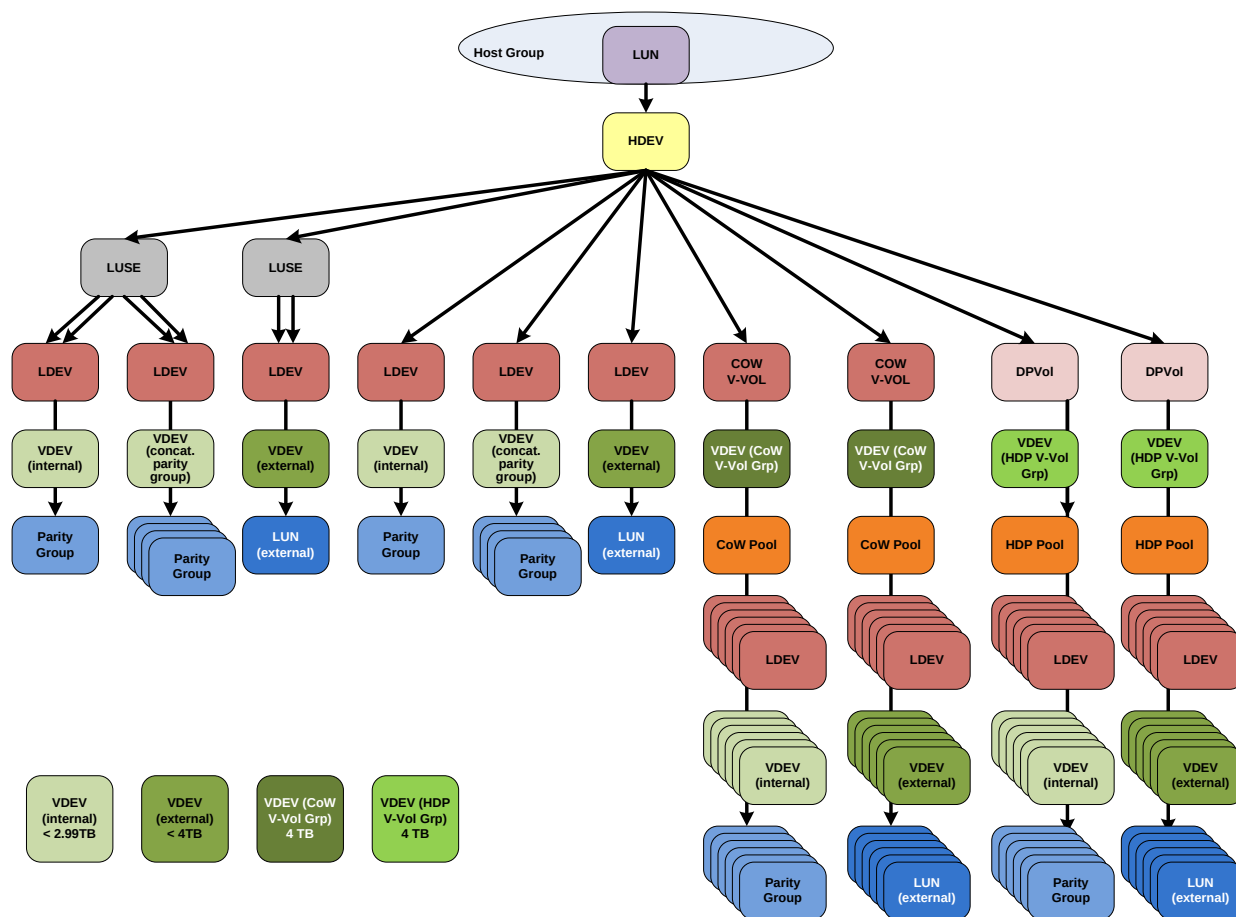


Figure 2. Alternate Overview of Logical Devices.



Note: On the VSP there no longer is the VDEV called the HDP V-Vol Group).

LDEVs are uniquely identified across the storage system by a six hexadecimal digit identifier usually written with a colon between each pair of hex digits, for example 00:00:01. This format is called the LDKC:CU:LDEV format, and thus the subfields with LDEV names are called the LDKC (Logical DKC) number, the CU (Control Unit) number, and the LDEV number (within the CU).

The maximum number of LDEVs that may be created within a VSP storage system is 64K (single chassis) or 128k (dual chassis). The LDKC field, the first 2 hex digits of an LDEV identifier, was added with the introduction of the VSP in order to provide for the ability at a later date to possibly increase the number of LDEVs per storage system above 64K.

Note that the term “LDEV” as used with HDS enterprise storage systems (VSP) to describe an internal logical volume, such as can be carved from a parity group, is somewhat confusingly called an “LU” or a “LUN” for HDS modular storage systems (AMS family). What is called an LU or LUN in HDS enterprise storage systems is called an “H-LUN” in modular storage systems. Sorry for the confusion.

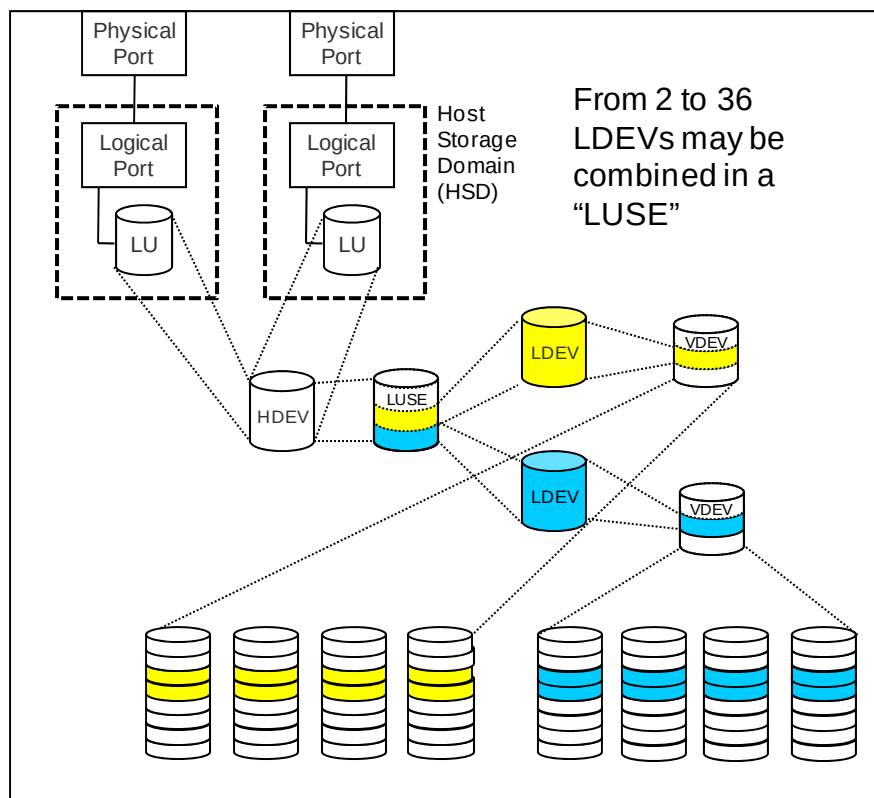
Note that a LUN, the way that a host “sees” an LDEV, emulates a physical drive. As such, accessing a LUN consists of reading or writing drive sectors comprised of 512 bytes. Thus the smallest amount of data that a host can read or write is one whole 512-byte sector. The size of a LUN or an LDEV is therefore a multiple of a 512-byte “block” (sector).

When creating an LDEV, if you specify the size in blocks (512-byte sectors) you will get an LDEV of that exact size. If you specify the size you want using any other unit, such as MB, there will be a rounding operation that will take the size you asked for and will round up to the next higher multiple of an internal allocation unit (logical cylinders of 15 logical tracks). Suffice it to say that if you want to make an LDEV of an exact size, specify the size in blocks (sectors).

HDEVs and LUNs or LUSE

Note in the diagram above that LDEVs are not directly presented to hosts as a Logical Unit (LU) or LUN, but rather there is an intermediate construct called an HDEV (Head LDEV). An HDEV may either map directly to a single LDEV, or it may be mapped to what is called a LUSE (Logical Unit Size Expansion). A LUSE is a combination of from 2 to 36 LDEVs:

Figure 3. Overview of LUSE Devices.



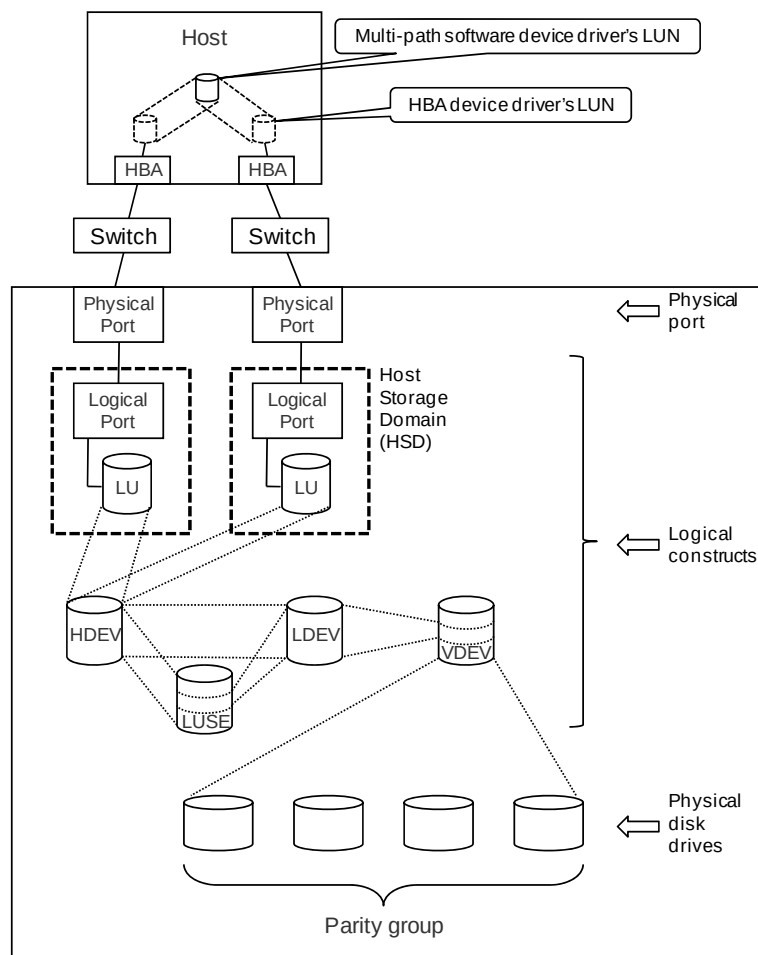
The only reason that one should make a LUSE is to create a LUN that is larger than the maximum available LDEV size. For internal LDEVs, the size of an LDEV is limited by the amount of formatted space in one parity group, or about 2.99 TB, whichever is smaller (more on this below when we talk about VDEVs). For other types of LDEV, the maximum size is 4 TB.

The maximum size of a LUSE, on the other hand, is 60 TB.

Note: The LDEVs within a LUSE are not interleaved or striped. LBAs (Logical Block Addresses, or sector numbers) for the LUSE start at the beginning of the first, or “head” LDEV, and then proceed to the end of each LDEV before starting on the next one. Thus when or writing sequentially, the performance will be limited by the performance of a single LDEV at a time.

Later on, we will talk about what are confusingly called “concatenated parity groups” which do not give the ability to make larger LDEVs, but which do interleave the RAID stripes of an LDEV across multiple parity groups.

Figure 4. Overview of LUSE Devices.



Going back to our original diagram for a moment, we can see that a single LDEV (storage system internal logical volume) may be exposed as multiple different LUNS to hosts. The diagram above shows a single host with two paths to an HDEV.

Granting a host access to an HDEV (LDEV or LUSE) is done by creating a Logical Unit (LU) within a Host Storage Domain. This mapping from a host to an HDEV is sometimes called a “path”. A path consists of three parts, the physical port (such as port 1A), the logical port, or Host Storage Domain within that physical port, and then the Logical Unit Number, or LUN within that HSD. Host Storage Domains may be identified by name, or by relative number within the physical port. Note that it is possible to map a particular LDEV as one LU number in one HSD and a different LU number in a different HSD.

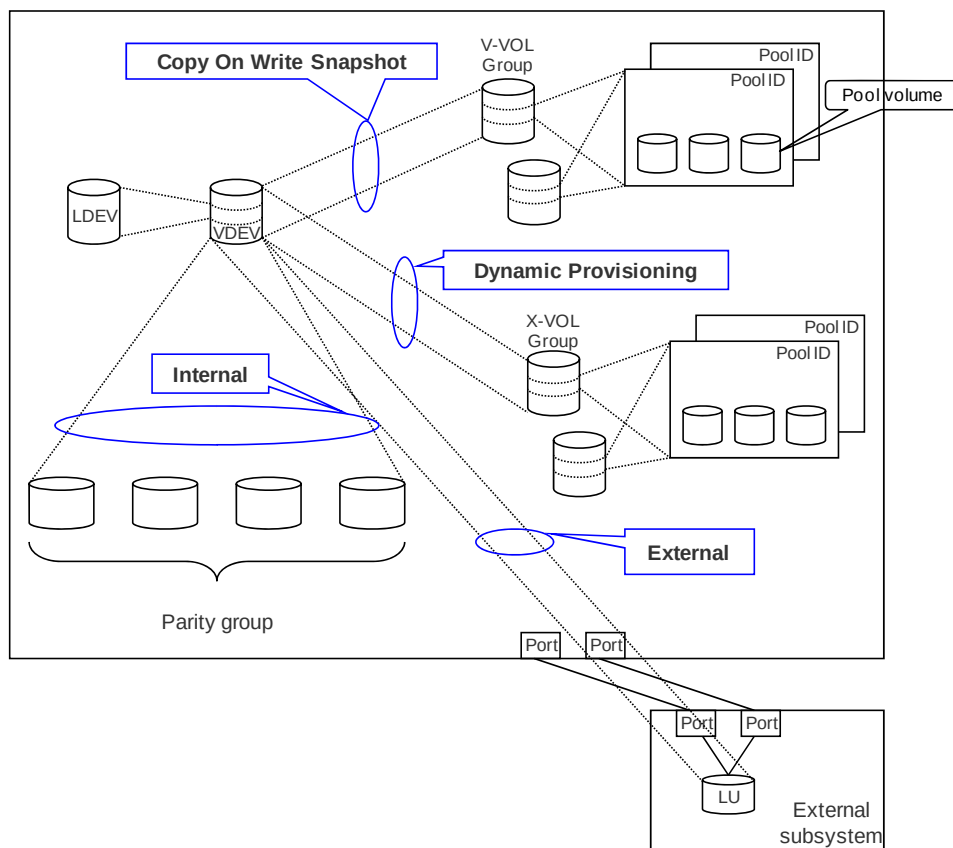
From the host’s perspective, the host doesn’t see the two-level structure of the physical port and HSD (logical port). When the HSD is configured, you tell the storage system which host(s) is to be mapped to each HSD. (A host can only be mapped to one HSD on a given port.) Thus the host “sees” a series of logical units (LUs) behind the port, where each LU is identified by its LU number.

The diagram above also illustrates how multiple paths from a host to an HDEV are implemented. The key is the use of multipath software in the host, such as Hitachi's HDLM (Hitachi Dynamic Link Manager). The multipath software is a device driver which presents a single logical device for applications to use. But when I/Os are directed to that logical device, the multipath software device driver receives those I/O operations and re-issues them to one or the other of the separate LUNs presented by each HBA's device driver. The HBA device driver's LUNs are hidden from applications, and are only used by the multipath driver software. It is only the multipath software that "knows" that the (hidden) LUNs from the two HBAs really lead to the same HDEV inside the storage system.

Correspondingly at the storage system end, when a host has multipath software running, the storage system is not aware of that fact (applies to USP and USP V generations – does not apply to the VSP or HUS VM). The storage system does not "see" I/O operations coming down separate paths as being related in any way. For this reason, the "extended round robin" host multipathing algorithm will produce better results for sequential I/O for active/active configurations. Extended round robin issues a series of I/O operations for given multi-path LUN over one of the paths before issuing the next series of operations over the other path. In this way, the storage system's "sequential detection" algorithm which is not aware of multipathing will still see sequential host I/O as sequential at the individual storage system port.

VDEVs

Recall that we said that LDEVs are "carved" from a logical container called a VDEV. There are four types of VDEV: internal, external, COW, and Dynamic Provisioning (DP) VDEVs.



Internal VDEV

The original type was the internal VDEV, which is a logical container that maps to the formatted space within a parity group, net of parity or mirror copies. Since LDEVs are carved from VDEVs, the size of the largest internal LDEV that can be created is one that entirely fills the VDEV that it is carved from.

There is a maximum size for internal VDEVs, which is about 2.999 TB. If there is more formatted space than this in the parity group, then as many maximum size VDEVs as will fit are first allocated, then any remaining space is allocated to one more VDEV.

You can see this if your parity group has more than 3 TB of space in it. In the following example, we have a RAID-5 7+1 parity group consisting of 750 GB drives.

No.	LDK:C:\LDEV	Emulation	Capacity
1	00:02:00	OPEN-V	3145663.500 MB

The effective capacity of each drive is not quite 750 GB, but just as a rough estimation the total formatted capacity of the parity group is $\sim 0.7 \text{ TB per drive} \times 7 \text{ drives} = \sim 4.9 \text{ TB}$. This is bigger than the max internal VDEV size, so you can see above that two VDEVs were created to represent the formatted space on this parity group. Here you can see that the parity group name is 1-1, and the VDEV names on this parity group are “1-1-(1)”, and “1-1-(2)”. That’s what those numbers in parentheses are at the end of a “parity group” name. They are the number of the VDEV within that parity group. Note that the LDEVs are within the VDEV, not the parity group.

While we are at it, parity groups have names like 1-1, 1-2, 3-1, and so on. These names have physical meaning, as they indicate the BED pair and physical drive location.

Thus the names for internal VDEVs are purely determined by the physical layout of the storage system.

External VDEV

Each External VDEV maps to a LUN on an external (virtualized) storage system. Thus each LUN on the external storage system corresponds to the equivalent of a parity group. It is possible to create an LDEV on an External VDEV that is the exact size in sectors of the external LUN. This makes it handy to do replication or migration.

The names for external VDEVs are in the form of either “E1-1” or “1-1 #”. Both forms of the name are equivalent, thus E 1-1 and 1-1 # both refer to the same external VDEV. In some places one form of the name is used, and in other places the other form is used.

Note that there is a separate name space for each of the four types of VDEV, so that even if you have an internal parity group called 1-1, you can create an external VDEV called E 1-1, and these are not the same.

The maximum size of an external VDEV is 60TB.

CoW VDEV

VDEVs for Copy on Write Snapshot (CoW) are usually called “V-VOL groups”.

Each CoW VDEV (CoW V-VOL group) is fixed in size at 4 TB.

CoW V-VOL groups have names that look like V1-1 or 1-1 V. These two forms are equivalent, with one form being used in some places and the other form the rest of the time.

A CoW V-VOL group does not map to any storage. It is purely a logical container from which CoW V-VOLs (CoW LDEVs) may be carved. It is these CoW V-VOLs each individually that are mapped to CoW pools.

V-VOL stands for Virtual VOLUME. This is because physical space is not provided for CoW V-VOLs when they are created. Instead, the LBA range representing extent of the V-VOL is divided up into 42 MB pages, where the pages are tracked in a page table. A physical location for a 42 MB page within the assigned CoW pool is only assigned when that page in the V-VOL is written to. Thus pages that have never been written to have no storage assigned. This comes in handy as CoW V-VOLs are point-in-time snapshot ShadowImage secondary volumes, and thus you only need space for whatever data has changed on the primary volume.

The underlying space in a CoW pool is provided by pool volumes, which are normal internal or external LDEVs exclusively assigned for duty as pool volumes. The space on the pool volume LDEVs is broken into 42 MB pages, each starting at a RAID stripe boundary. Thus one page consists of a series of multiple RAID stripes.

A note on pool volumes - when an LDEV is assigned as CoW or Hitachi Dynamic Provisioning pool volume, the suffix “P” is shown on the name of the parity group that the pool volume LDEV is on.

For example, if a normal LDEV 00:12:34 which is on parity group “1-1” is assigned as a CoW or Hitachi Dynamic Provisioning pool volume, from that point on, LDEV 00:12:34 will show as being on parity group “1-1 P”. In this case, the “P” is really a marker to indicate special treatment as a pool volume, not that there is a parity group whose name is “1-1 P”.

You can see this using the SVP configuration printout tool.

Dynamic Provisioning VDEV

DP VDEVs are also called “V-VOL groups”, although their names are written in the form X1-1 or 1-1 X. Dynamic provisioning is exactly like where a COW secondary V-VOL has been mounted to a host for host access, and where the CoW primary is all zeros.

In fact, although any one pool must be either a CoW pool or a DP pool, there is only one name space for COW/DP pools. The maximum number of pools within a storage system is 128.

Dynamic Provisioning V-VOLs are LDEVs carved from a DP Pool and are usually called “DPVOLs” for short.

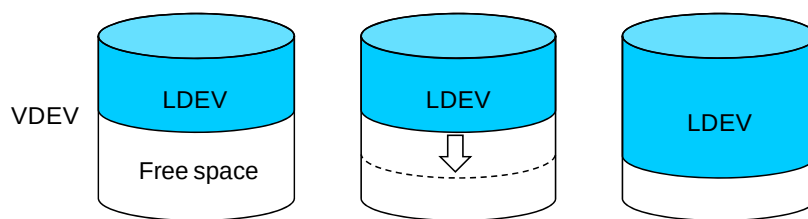
The mechanism is the same as for CoW. When a host writes to a 42-MB page on a DPVOL that has never been written to before, a page is allocated from the pool, and from then on whenever the host reads or writes from that page, the access is directed to the assigned physical page in the pool.

DP VOL groups (VDEVs) are limited in size to 60TB.

When you look at the performance statistics for Dynamic Provisioning, what you see is that the DPVOLs (the virtual volumes) are the ones that the hosts access, so they have front end I/O statistics only. On the other hand, pool volumes only have back end I/O statistics, as they are the ones that to which the actual back end I/O is performed.

The maximum number of LDEVs within a VSP is 65,280. This is also the maximum number of VDEVs in a storage system, and therefore it is possible to place each DPVOL by itself at the “top” of its own V-VOL group (DP VDEV).

In general, since LDEVs are carved from VDEVs, if there is contiguous unused space immediately following the LDEV within the VDEV it is possible to expand the size of an existing LDEV into that free space, as shown in the following diagram:



An LDEV may be grown “in place” by extending into immediately following (contiguous) free space within the VDEV

During this expansion, any existing data in the LDEV stays “in place”, and all that happens is that the LDEV gets more space at the end. Note that appropriate operating system support is required for the host to “notice” that the volume got bigger, and adjust the partition table on volume accordingly. Then an existing partition may be dynamically expanded, again if the operating system and file system support that.

Layout of internal VDEVs on the parity group

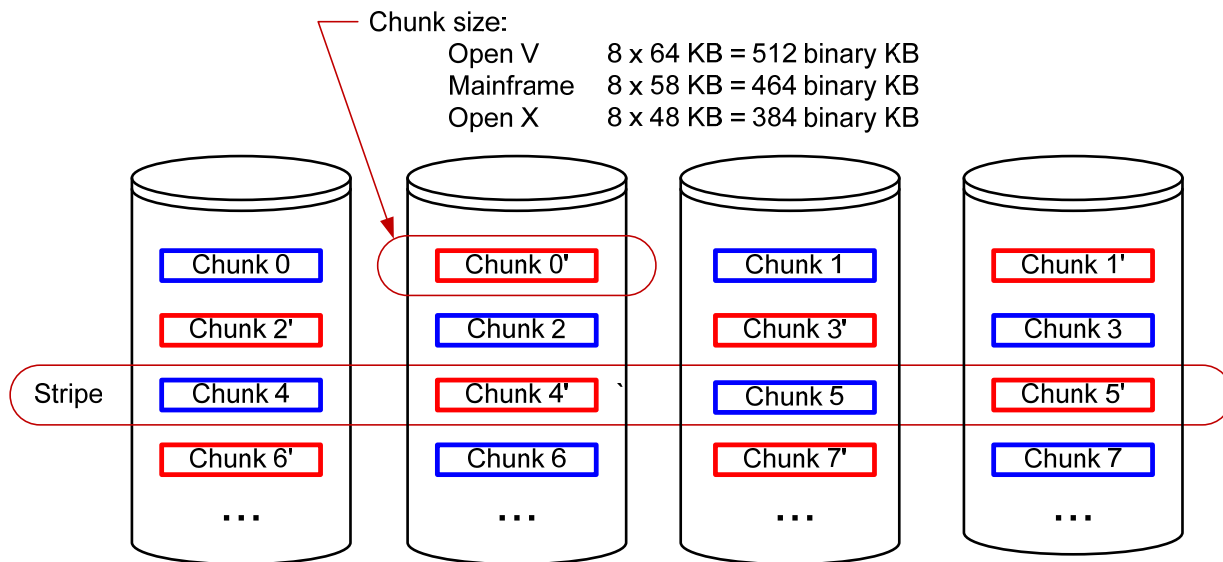
Recall that the VDEV is the logical container from which LDEVs are carved, and that internal VDEVs represent the formatted space within a parity group.

This abstraction of a VDEV as a logical container allows operations that deal with LDEVs to be implemented largely independently of the type of VDEV, and thus LDEVs work the same way whether they are internal, external, CoW V-VOL, or DPVOL.

Thus it’s in the internal type of VDEV where the mapping happens to the formatted space within a parity group.

In the next two diagrams, we see the VDEV layout on first RAID-1 (2D+2D) and then RAID-5 (3D+1P) parity group types.

VSP RAID-1 2+2 VDEV chunk layout



Thus the VDEV is mapped to “chunks” on the drives. In the Hitachi Virtual Storage Platform (VSP), the chunk size varies only with the emulation type, and not by the RAID level or number of drives in the parity group.

Note that the chunks are relatively big. About a hundred 4KB blocks fit within one chunk.

For random I/O, only the data requested by the host is read or written. If the host reads a 4 KB block, only 4KB is read from the drive, not an entire chunk.

The time it takes to do a random I/O is composed of over 99% mechanical positioning and less than 1% data transfer. This is because a block size in the 4KB to 8KB range represents on the order of only 1% of a drive physical track size, and mechanical positioning (seek + latency) takes on average more than one full drive rotation.

That’s why for random I/O workloads, all we care about are the drive seek times and RPM.

The chunk size is chosen so that the amount of time it takes to transfer a whole chunk (the size of the chunk divided by the read/write head sustained data transfer rate) is of the same order of magnitude as the mechanical positioning time, that is, seek and latency time.

It’s designed this way because sequential I/O operations read or write an entire chunk at a time. With having such big chunks, very roughly one half of the drive busy time when reading or writing sequentially is mechanical positioning and one half is data transfer.

This makes sequential transfer relatively productive.

It also makes the I/O service time to read or write a chunk about twice as long in time as a small-block random I/O. The mechanical positioning is the same for both, but for large block sequential I/O after

the mechanical positioning is complete it takes roughly one drive rotation worth of time to transfer the chunk.

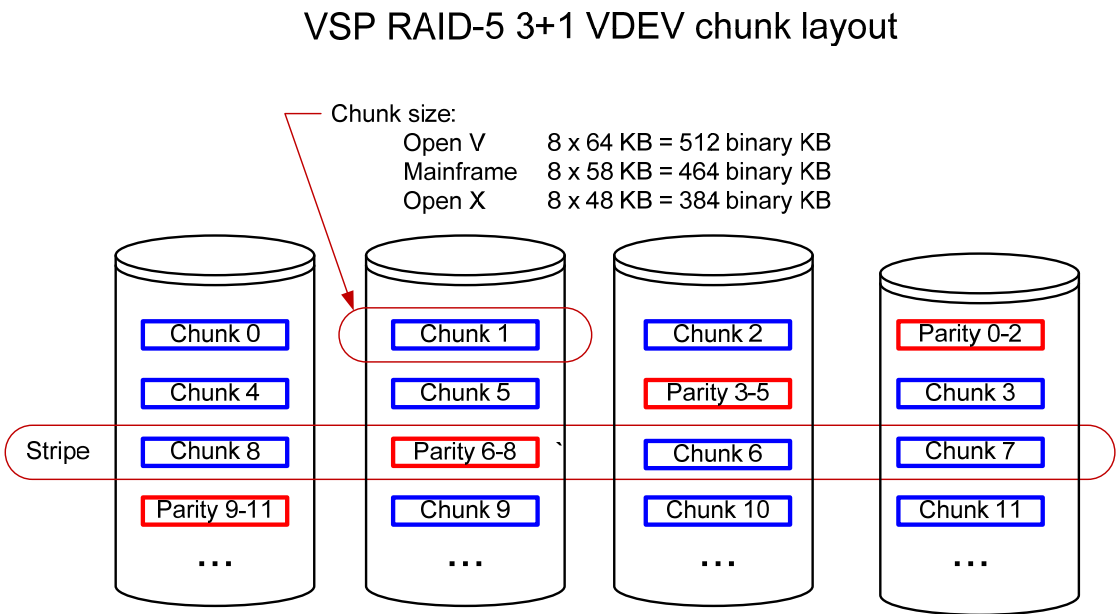
You might ask why not make the chunks even bigger? If transferring a 512KB chunk represents almost ½ of the total I/O service time, then why not make the chunks 1MB? That way you would get a 50% throughput increase for sequential I/O, because then almost 2/3 of the drive busy time during a sequential I/O would be data transfer.

The reason is that if a random read miss occurs while a sequential I/O (pre-fetch stage or write destage) is in progress, the host has to wait. Sequential I/O operations already take about twice as long as random I/Os, and if we made the chunks even bigger the host would have to wait even longer if a random read miss occurs during a sequential I/O. Thus there's a trade-off as a larger chunk size would increase sequential throughput at the expense of adversely impacting random read performance.

Note also in the RAID-1 VDEV chunk layout diagram that the primary chunks (shown colored blue) and the mirror copy chunks (colored red) are in alternating positions from row to row. This is done because the mirror copies are normally “write only” as a backup. All read operations are directed to the primary copy unless the drive containing the primary copy has failed.

Thus the effect of the alternating primary and mirror chunks in the layout from row to row in RAID-1 tends to balance the number of read operations directed to each drive within the parity group.

Let's take a look at the chunk layout for a RAID-5 (3D+1P) VDEV in the VSP:



You might ask “Why does the 2nd row have the chunks numbered in the order 4, 5, 3? This is confusing.” The reason “why” is that, in the VSP, sequential reads will read a chunk from each drive in the parity group in parallel. The layout above enables chunks 1,2,3,4 to be read at once. In other words, sequential reads do not read one parity group stripe at a time. However, for sequential writes one stripe at a time (in this case 3 data blocks and one parity block), again performing one I/O to each drive in the parity group at once.

Note: The HDS AMS 2000 modular family of storage systems uses a different RAID-5 chunk layout scheme which is more what you might have expected. The analogous diagram for 3+1 for a modular storage system has in row 2 from left to right chunks 3, 4, parity 3-5, and chunk 5.

We said that the abstraction of a VDEV as a logical container allows operations that deal with LDEVs to be implemented largely independently of the type of VDEV. One small dependency on the type of VDEV is that LDEVs always start on a RAID stripe boundary. That is, the first chunk of an internal LDEV is always the first data chunk of a RAID stripe, or parity group “row”.

This is done to make it easy to calculate the physical location of where a particular LBA is stored. Each LDEV has an attribute which is the LBA address of where that LDEV starts on every drive in the parity group. Thus you can divide an LDEV LBA by the RAID stripe size (data-only stripe size, exclusive of parity or mirror copies), and the quotient gives you the relative stripe offset in chunks on the drive, relative to where the LDEV starts on each drive, and the remainder tells you where you are within the stripe and therefore which drive contains that LDEV LBA. (The number of the stripe on the drive also tells you which chunks on that stripe are the data chunks and which are the mirror copy / parity chunk(s).)

Appendix 5. Concatenated Parity Groups

The VSP family has exactly the following 4 base Parity Group types:

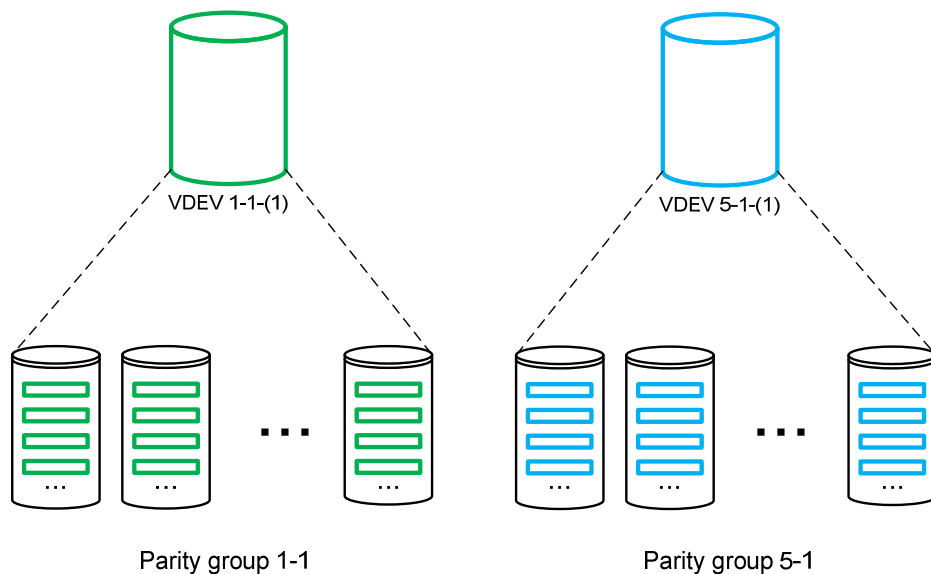
- RAID-1 (2D+2D)
- RAID-5 (3D+1P)
- RAID-5 (7D+1P)
- RAID-6 (6D+2P)

Note that there is no “4+4” as a base Parity Group type.

The following conceptual diagram, showing two Parity Groups that are not concatenated, will be used as the starting point to explain two-way concatenation:

Two separate parity groups, without concatenation

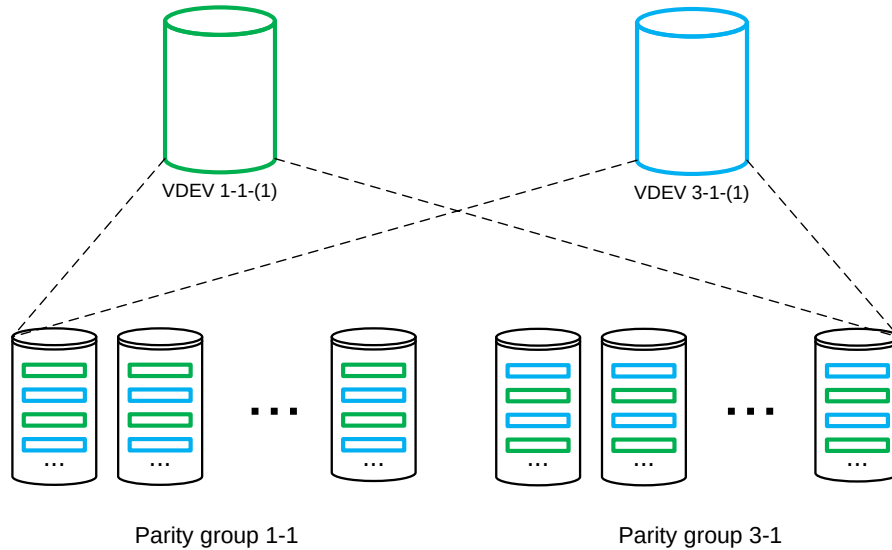
NOTE: The VDEV actually maps to the formatted (data) chunks in the parity group. This simplified diagram for both RAID-1 and RAID-5 shows all the chunks, both data and mirror copy / parity.



Concatenated Parity Groups are derived from the base Parity Group types by interleaving their VDEVs. The following diagram shows a two-way concatenation or interleaves:

Two-way parity group concatenation (VDEV interleave)

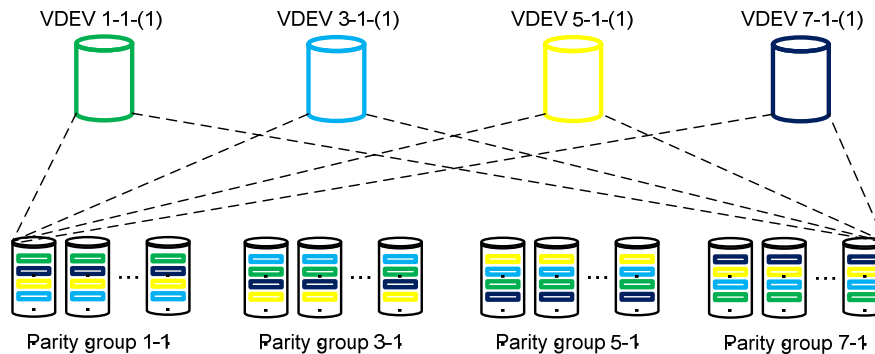
NOTE: Concatenation does not create one big VDEV for both parity groups. Each VDEV has its first stripe on the parity group that it is named after. Thereafter, RAID stripes for a given VDEV alternate between the parity groups in the concatenated set.



There are two types of two-way interleaved Parity Groups.

- The two-way interleave of RAID-1 (2D+2D) is what we usually call '4+4'.
 - Note that there is a VDEV for each Parity Group that is the same size as the VDEV would have been without concatenation. Thus the biggest LDEV you can make on a 4+4 is the same size as the biggest LDEV you can make on a 2+2.
- The two-way interleave of RAID-5 (7D+1P) doesn't have a shorthand name, but if you write $2 \times (7+1)$ this is clear. Sometimes people write "14+2" – if they are writing about the VSP family you know that a 14+2 could only be $2 \times (7+1)$.

Four-way parity group concatenation (VDEV interleave)



There is one type of four-way interleaved Parity Group.

- The four way interleave of RAID-5 7+1 may be clearly written as $4 \times (7+1)$. If you see someone write $28+4$, this is what they must be talking about.

Yes, the name is confusing.

You will sometimes see the terms “VDEV disperse” or “VDEV interleave”. These are equivalent to the official name “concatenated Parity Groups”. Sometimes people will talk about LDEV striping, and here it is possible that they are talking about concatenated Parity Groups, because the LDEVs that are carved from a VDEV that is part of a concatenated Parity Group set are indeed striped across the Parity Groups that have been concatenated together.

The term “concatenated” is unfortunate, as many native English speakers would assume that this refers to something more like a LUSE. A colloquial English expression often used to describe LUSE is “spill and fill”, because in a LUSE, reading or writing sequentially through the LUSE, one proceeds through each constituent LDEV in its entirety before moving to the next. Thus LUSE does not improve sequential throughput.

Advantages of concatenated Parity Groups

- Individual LDEVs are striped across the set of concatenated Parity Groups. This means that sequential performance is increased, as when staging or destaging sequential data, an I/O to read or write a chunk may be issued to all the drives in the concatenated Parity Group set at the same time.
- Random I/O operations are effectively dispersed across the set of concatenated Parity Groups. If small-block random I/O operations are clustered to within up to a hundred blocks or so, they will likely go to the single drive that contains the chunk (512 KB for Open-V) that contains those small blocks. This is true for any Parity Group type. However, the advantage of concatenated Parity Groups is that if random I/O is clustered within tens or hundreds of MB, these I/Os will be dispersed across the whole set of concatenated Parity Groups. In the case of a LUSE volume, random I/O clustered within 10s or 100s of megabytes will likely be concentrated on a single LDEV that is on a single Parity Group.

- Concatenated Parity Groups reduce to some degree individual Parity Group “hot spots”. The total activity across all the LDEVs on a set of concatenated Parity Groups is distributed more or less evenly across that set of Parity Groups. Thus this “averages out” Parity Group utilization across the concatenated set, and thus reduces average Parity Group busy. Reduced “Parity Group busy” yields better overall average response and offers the opportunity to increase overall throughput.

Note that Dynamic Provisioning offers the benefits of I/O activity dispersal across much larger numbers of Parity Groups.

Disadvantages of concatenated Parity Groups:

- Since I/O activity is dispersed over a concatenated set of Parity Groups, all LDEVs carved from VDEVs in the set are sharing the same performance characteristics. Thus there is less “performance isolation”. Normally dispersing activity is a good thing, but it should be pointed out that any of the LDEVs in the combined set have access to the full bandwidth of the drives in the set, and thus high-activity LDEVs will make the entire concatenated set of drives busy, and thus reduce the performance of the other LDEVs sharing those drives.
- Should a drive fail, performance could be affected for a larger set of LDEVs.
- Most people expect that when you “concatenate” Parity Groups together, you will be able to make bigger LDEVs. As we saw, concatenation does not alter the number or size of the VDEVs that would otherwise be assigned to the same Parity Groups if they were not concatenated together. To fix this problem, if you want the performance characteristics of LDEVs that are striped across a concatenated set of Parity Groups, but you also want a LUN that is as big as the combined formatted size of the Parity Groups in the set, then carve a single maximum-size LDEV out of each VDEV belonging to a set of concatenated Parity Groups, and LUSE these LDEVs together to make one LUN.

Using LDEVs on concatenated Parity Groups as Dynamic Provisioning pool volumes

Recall that DPVOLs are divided up into 42 MB pages, and that each individual page when it has been written to is assigned a location in the pool.

42 MB is “fairly big”, in the sense that the biggest RAID stripe is an Open-V RAID-5 (7D+1P) stripe which has 7 data blocks of 512 KB each for a total of 3.5 MB of data.

What we thus see is that each 42 MB DP/CoW page is comprised at the Parity Group level of a series of RAID stripes. Each page starts on a RAID stripe boundary. The 42 MB size of a page was picked so as not to waste any space in any of the base Parity Group types. One 42 MB page is comprised of:

- 42 stripes of a 2+2, where the stripe size is $2 \times 512 \text{ KB} = 1 \text{ MB}$.
- 28 stripes of a 3+1, where the stripe size is $3 \times 512 \text{ KB} = 1.5 \text{ MB}$.
- 12 stripes of a 7+1, where the stripe size is $7 \times 512 \text{ KB} = 3.5 \text{ MB}$.
- 14 stripes of a 6+2, where the stripe size is $6 \times 512 \text{ KB} = 3 \text{ MB}$.

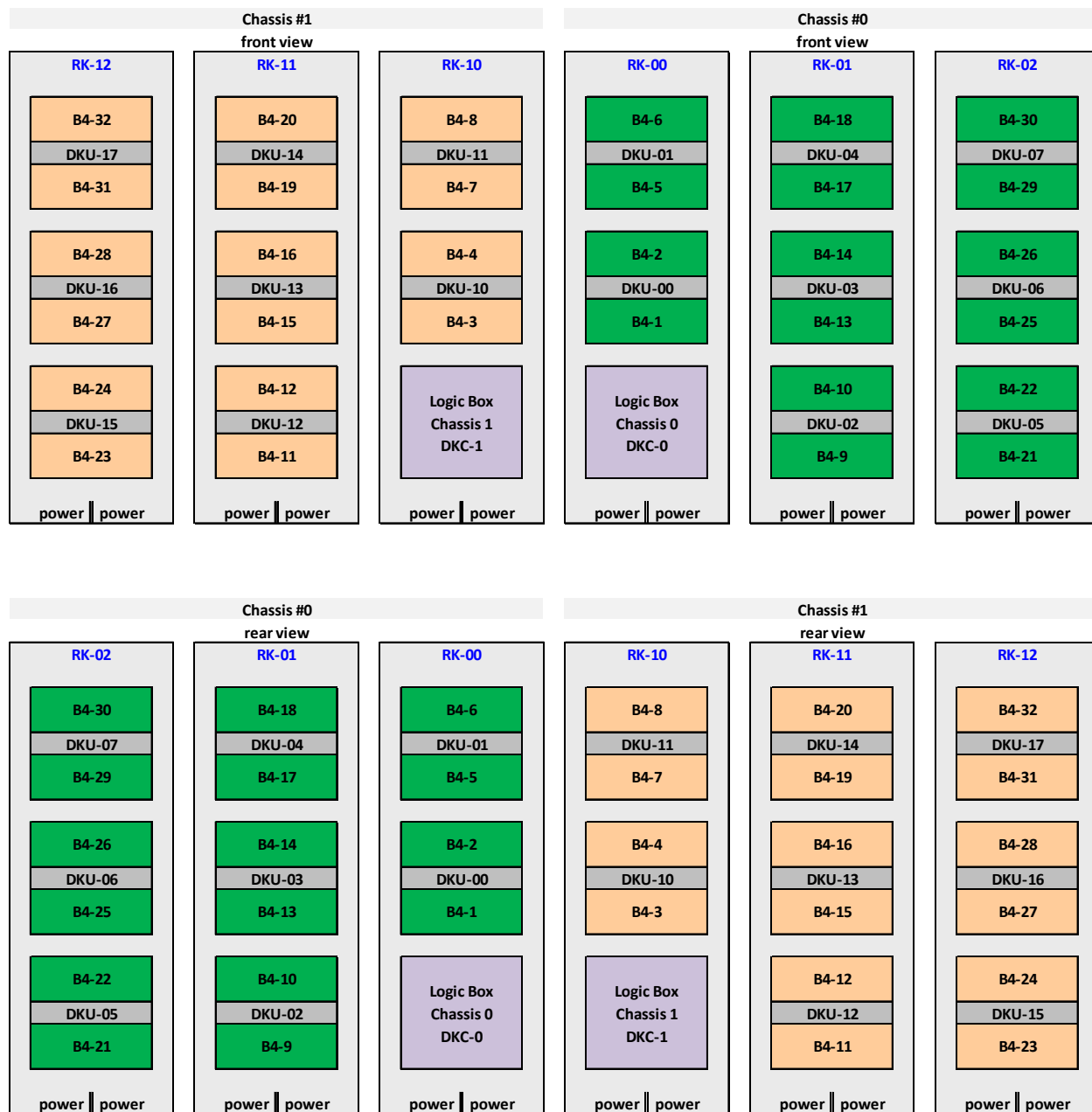
We have not specifically tested the performance of LDEVs on concatenated Parity Groups used as DP/CoW pool volumes. However, there is a theoretical argument that says this might improve

sequential performance. Since sequential performance is higher for normal LDEVs on concatenated Parity Groups due to the characteristic of reading / writing a chunk to all the drives in the concatenated set at once, (reading / writing one “round” of interleaved RAID stripes at ones) it would be worth trying to see if this also applies to sequential processing of DP/CoW pages. This may or may not provide a benefit in the context of DP/CoW sequential processing. This document may be revised in future to clarify this point, but for the time being this remains an open question.

Summary of the characteristics of concatenated Parity Groups (VDEV interleave):

- The size of the VDEVs is exactly the same as the size of VDEVs on similar Parity Group types that are not interleaved.
- No change to the maximum size of an LDEV.
- It is not possible to allocate a single LDEV that consumes all the space in all the Parity Groups being concatenated (interleaved) together. Instead, create a maximum-size LDEV on each VDEV associated with the set of concatenated Parity Groups and LUSE these LDEVs together.
- The I/O activity of each LDEV carved out of a concatenated Parity Group VDEV maps to the underlying RAID stripes comprising the VDEV, which means that the I/O activity spreads across all the drives in the concatenation set.
- Sequential I/O throughput is improved because you can read or write one chunk to each of the drives in the concatenated set in parallel.
- Random I/O operations benefit from having an LDEV’s I/O operations distributed across a larger number of drives.
- RAID group performance isolation happens at the level of the concatenated set, as all the LDEVs in all the VDEVs in the concatenation set share the same drives.
- In addition to the base Parity Group types 2+2, 3+1, 7+1, and 6+2, concatenated Parity Groups yield the types 2x(2+2), which is commonly called 4+4, as well as 2x(7+1) and 4x(7+1).

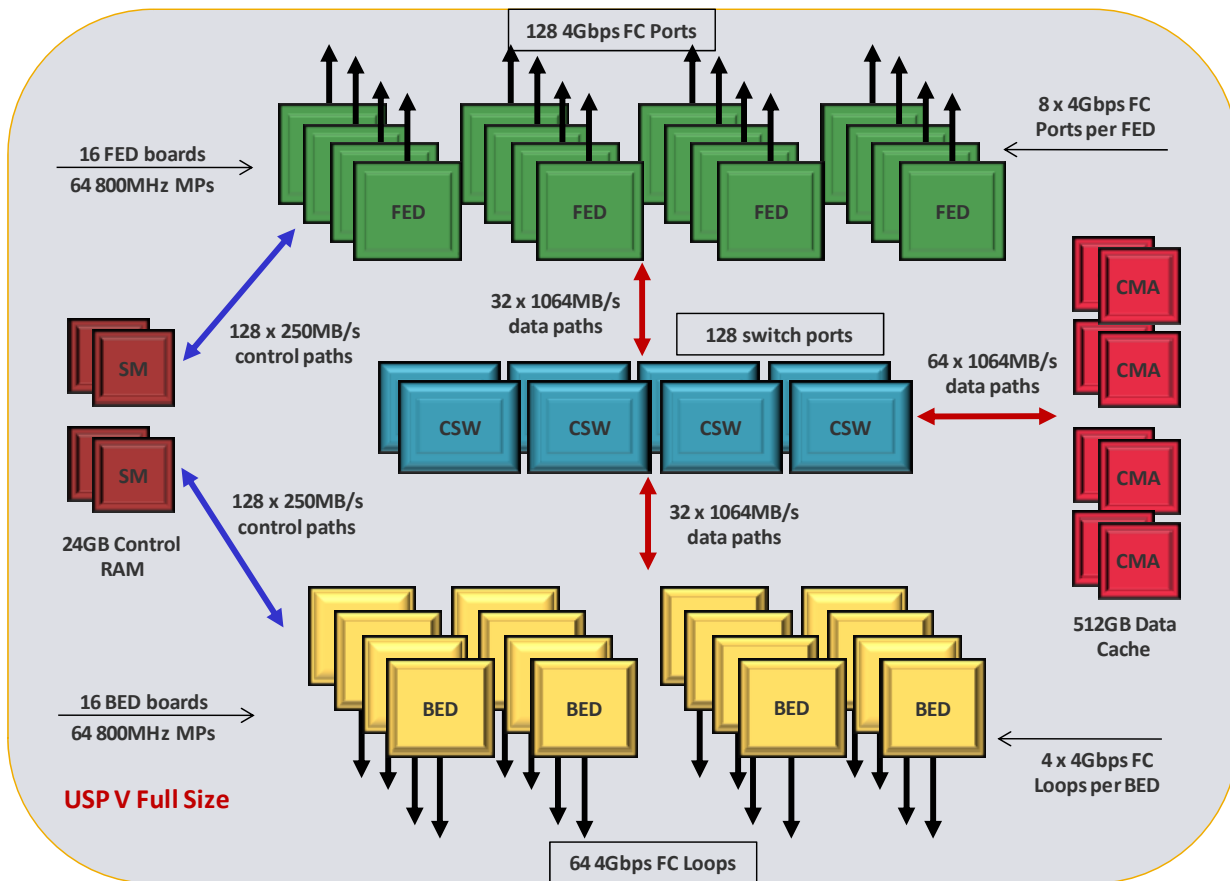
Appendix 6. VSP B4 Maps – Parity Groups Names



Appendix 7. USP V Architecture Overview

This figure shows the board level organization of the fully configured USP V storage system in the style used in this paper. The SM boards are Shared Memory, the CMA boards are cache, and the CSW boards are the PCI-X cache switches.

Some of the BED boards (up to 12) can be replaced by FED boards for a USP V that would primarily be used for virtualization. This would provide a maximum of 192 host port, with only 16 FC drive loops.



Appendix 8. VSP Hardware Maintenance Manual, page THEORY02-210 Explained

One of the factory figures that can cause some people confusion is this one. Keep in mind that this only shows the front side of one DKU (either the LFF or SFF type). The SAS boundary term used here merely describes the association of the 4 or 8 SAS links from one or two BED features (over 2 or 4 SAS 2W cables) to the pair of switches connecting that pair of HDUs. Those 2 or 4 cables are daisy-chained to the next DKUs (up to 7) to attach to that same pair of HDUs in each. So the SAS boundary is really the set of 1 to 8 such HDU pairs of switches.

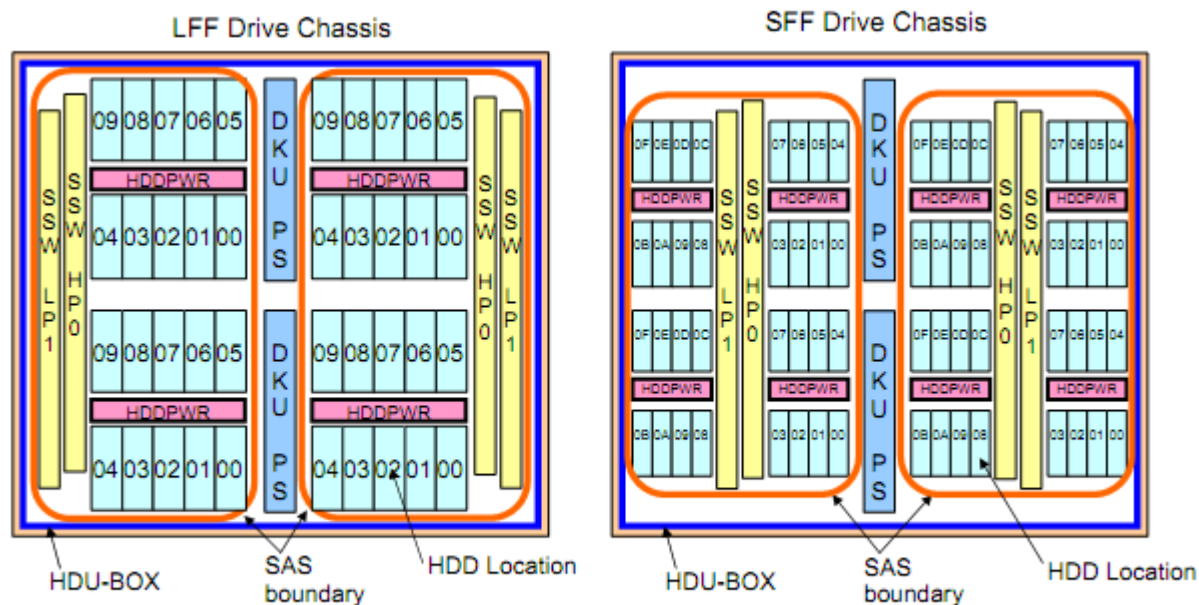


Fig. 2.2.3-4 Structures of Drive Chassis

In the next factory figure, only one BED feature is shown, and each line is a 2W cable. So each switch in that DKU has two SAS links connected to it. If this were a chassis with two BED features, then there would be two such lines per switch in this figure, or four SAS links. Each yellow SSW box is a SAS switch shared by two HDUs (those in a vertical stack in that DKU). There are two such switches associated with each pair of HDUs (this is a "SAS Boundary"), and as such they are controlling either 20 (LFF DKU) or 32 (SFF DKU) drives. Each blue "HDD" represents one HDU. If these were numbered as HDD-1 to HDD-8 (from left to right), then four-disk Parity Groups (3+1 and 2+2) will have one drive each in HDD-1, HDD-3, HDD-5, and HDD-7, OR HDD-2, HDD-4, HDD-6, and HDD-8. The drives for any 8-disk Parity Group (7+1, 4+4, and 6+2) will each come from one of these eight HDUs.

In each VSP chassis, this means that each 4-disk or 8-disk Parity Group in a DKU will be supported by the same 16 SAS links (single BED) or 32 SAS links (dual BED). Note that drives need to be evenly dispersed across the HDUs (for partially populated DKUs) in order to bring all SAS links to bear within that particular DKU. In other words, if this was an SFF DKU, and there were only 64 drives being installed, 8 of them should go into each HDU. But, in order to build 8-disk Parity Groups, this approach would be required anyway.

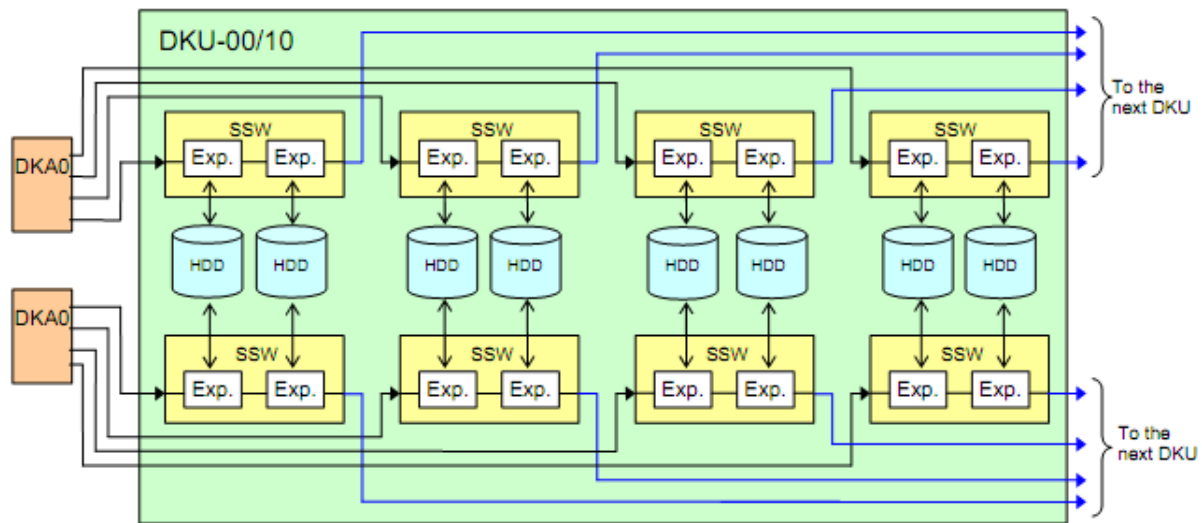
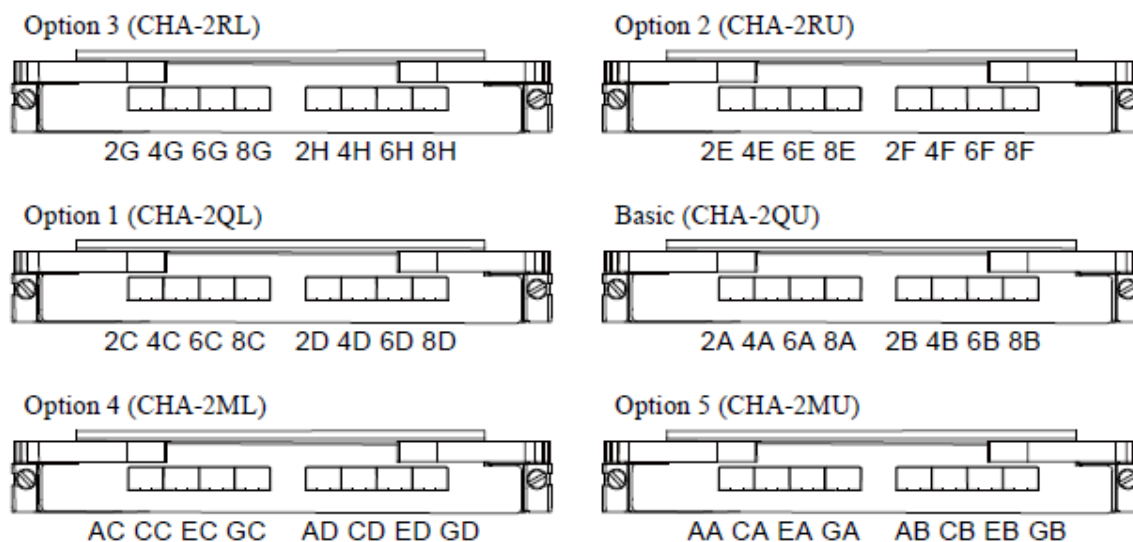


Fig. 2.2.3-5 Connections Diagram of Drive Chassis

Appendix 9. VSP Open System FED Port Names

These images are taken from the Maintenance Manual and shown here for convenience.

CHA PCB (Cluster 2)



CHA PCB (Cluster 1)

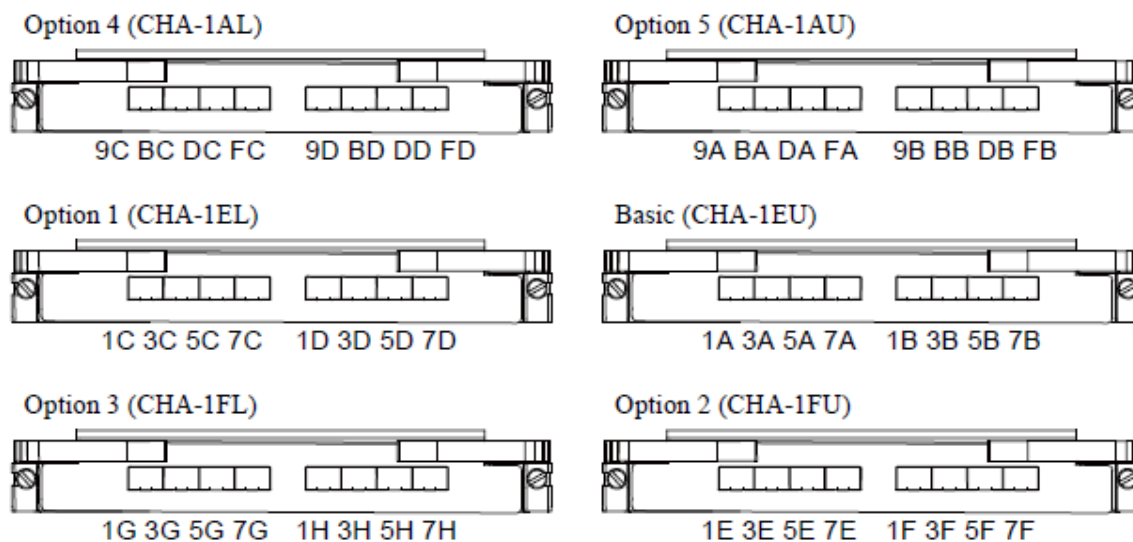
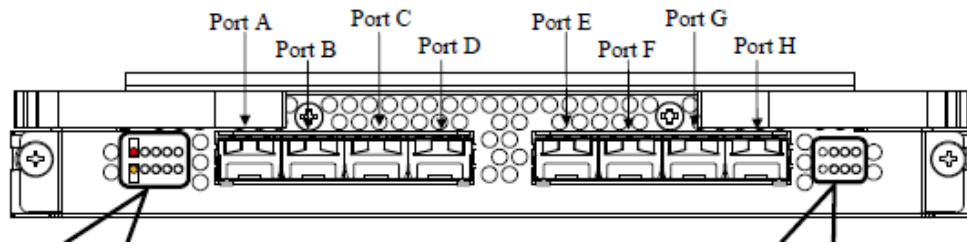


Fig. 4.1-3 Port Number of Fibre 8-port CHA PCB

Rear View of DKC

CHA PCB



Appendix 10. PCI Express Details

The VSP Grid interconnection system uses PCI Express (PCIe) 1.0 serial busses for all of its internal connections instead of the PCI-X parallel busses (533MB/sec half duplex, either send or receive) that were used on the USP and USPV models. There are several sub-versions of each PCIe level (such as 1.1, 1.2, 1.3) so I will use the notation of 1.x (and such) here as my general reference.

PCIe busses provide a set of “lanes” that are combined together to work in parallel in what is termed a “PCIe link”. The number of lanes present in a link is noted as x1, x4, x8, and x16 in most discussions. [The specification allows for x32, but that isn’t seen outside of a lab.] Each lane has four wires: one differential pair for send and one differential pair for receive. Each pair moves 1-bit of data in a serial connection in one direction. The send and receive pairs operate independently of one another (full duplex) but share the same clock. Understand that the peak data rate of a lane is usually expressed as the aggregate of both directions working simultaneously even though the effective rate for 100% send or 100% receive is only half of that rate.

PCIe (1.x and 2.x) uses a 10-bit protocol (like Fibre Channel and Ethernet) to move data. This means that every 8 bits of data have 2 bits of parity. There is also a header for each packet of data on a link. This means that only about 80% of the aggregate protocol rate is used to move data (as is the case with FC and Ethernet). PCIe 3.x was redesigned to use a far more efficient 128bit/130bit packet which required very different sending and receiving hardware. So it can be up to 98.5% efficient.

PCIe is typically implemented as a 1-lane, a 4-lane, or an 8-lane bus. The bus controller will send data packets across the link (the number of available lanes) in a parallel byte manner. So x1 sends a single byte (serially), while x4 sends four bytes and the x8 sends 8 bytes in parallel. Each link can also receive at the same rates at the same time. The sender logic breaks down the data stream from the processor into these parallel streams and the receiver logic recombines them into a packet.

Looking at the table below, note that each x1 PCIe 1.x link supports a 250MB/sec send plus a 250MB/sec receive data rate which (theoretically) provides a peak data rate of 500MB/sec. Each x4 1.x link supports a 1GB/sec send plus a 1GB/sec receive data rate. Each x8 1.0 link supports a 2GBps send plus a 2GBps receive data rate.

All of the boards used in the VSP have four x4 links per board, except for the FED boards which use a pair of x4 links. This means that the FED boards can support an aggregate data rate (under heavy sustained usage) to and from the Grid Switches of 2GB/sec send and 2GB/sec receive, or an aggregate data rate of 4GB/sec. The other boards have four such links and can support an aggregate data rate of 8GB/sec (all four busses having sustained, simultaneous send plus receive traffic).

PCIe 1.0 uses a 2500MHz clock whereas PCIe 2.0 uses a 5000MHz clock. This is the only significant difference between them (there may be some signal handling differences in the chips). All of the performance numbers for 2.0 are double those listed above for the 1.0 bus. PCIe 3.x uses an 8000MHz clock with very different chips, and due to the greatly increased efficiency of the data encoders can achieve double the rates of PCIe 2.x even though the clocking rate didn’t double.

The table below summarizes all bus protocols for PCI, Fibre Channel, SCSI, and USB.

Protocol	Common	Performance	Bus	Bus	Aggregate	Aggregate
Type	Name	ratios to PCIe 2.0 x4	Width	Clock	Protocol Rate	Data Rate
PCI 32-33	PCI	0.03 x	32 bit	33 MHz	132 MB/s	106 MB/s
PCI 32-66	PCI	0.1 x	32 bit	66 MHz	264 MB/s	211 MB/s
PCI 64-66	PCI	0.1 x	64 bit	66 MHz	528 MB/s	422 MB/s
PCIx 100	PCI enhanced	0.1 x	32 bit	100 MHz	400 MB/s	320 MB/s
PCIx 100-64	PCI enhanced	0.2 x	64 bit	100 MHz	800 MB/s	640 MB/s
PCIx 133	PCI enhanced	0.1 x	32 bit	133 MHz	532 MB/s	426 MB/s
PCIx 133-64	PCI enhanced	0.2 x	64 bit	133 MHz	1064 MB/s	851 MB/s
PCIx 266-64	PCI enhanced	0.4 x	64 bit	266 MHz	2128 MB/s	1702 MB/s
PCIx 533-64	PCI enhanced	0.9 x	64 bit	533 MHz	4264 MB/s	3411 MB/s
PCIe x1 1.x	PCI Express 1-lane	0.1 x	1 bit	2500 GT/s	625 Mbit/s	500 MB/s
PCIe x4 1.x	PCI Express 4-lane	0.5 x	4 bit	2500 GT/s	2500 Mbit/s	2000 MB/s
PCIe x8 1.x	PCI Express 8-lane	1.0 x	8 bit	2500 GT/s	5000 Mbit/s	4000 MB/s
PCIe x16 1.x	PCI Express 16-lane	2.0 x	16 bit	2500 GT/s	10000 Mbit/s	8000 MB/s
PCIe x1 2.x	PCI Express 1-lane	0.3 x	1 bit	5000 GT/s	1250 Mbit/s	1000 MB/s
PCIe x4 2.x	PCI Express 4-lane	1.0 x	4 bit	5000 GT/s	5000 Mbit/s	4000 MB/s
PCIe x8 2.x	PCI Express 8-lane	2.0 x	8 bit	5000 GT/s	10000 Mbit/s	8000 MB/s
PCIe x16 2.x	PCI Express 16-lane	4.0 x	16 bit	5000 GT/s	20000 Mbit/s	16000 MB/s
PCIe x1 3.x	PCI Express 1-lane	0.5 x	1 bit	8000 GT/s	2000 Mbit/s	1970 MB/s
PCIe x4 3.x	PCI Express 4-lane	2.0 x	4 bit	8000 GT/s	8000 Mbit/s	7880 MB/s
PCIe x8 3.x	PCI Express 8-lane	3.9 x	8 bit	8000 GT/s	16000 Mbit/s	15760 MB/s
PCIe x16 3.x	PCI Express 16-lane	7.9 x	16 bit	8000 GT/s	32000 Mbit/s	31520 MB/s
FC-1	1Gb FC	0.024 x	1 bit	1000 MHz	120 MB/s	96 MB/s
FC-2	2Gb FC	0.048 x	1 bit	2000 MHz	240 MB/s	192 MB/s
FC-4	4Gb FC	0.096 x	1 bit	4000 MHz	480 MB/s	384 MB/s
FC-8	8Gb FC	0.192 x	1 bit	8000 MHz	960 MB/s	768 MB/s
UltraSCSI 80	SCSI	0.016 x	16 bit	40 MHz	80 MB/s	64 MB/s
UltraSCSI 160	SCSI	0.032 x	16 bit	40 MHz	160 MB/s	128 MB/s
UltraSCSI 320	SCSI	0.064 x	16 bit	80 MHz	320 MB/s	256 MB/s
USB-1.2	USB	0.000 x	1 bit	12 MHz	2 MB/s	1 MB/s
USB-2.1	USB2	0.012 x	1 bit	480 MHz	60 MB/s	48 MB/s
USB-3.0	USB3	0.125 x	1 bit	5000 MHz	625 MB/s	500 MB/s
IEEE-1394	firewire	0.010 x	1 bit	400 MHz	50 MB/s	40 MB/s

NOTE: **GT/s** means **Giga-transfers per second**, which is how most things (CPU front-side busses, graphics CPUs, PCIe) are rated now. I switched to using GT/s in this table since that is how all PCIe 2.x and later documents will show it. For each clock cycle, all of the data in a transfer register is moved. These registers vary a lot in sizes (from 64-bit to 512-bits) and thus the transfer rates, so it is more meaningful to use the GT/s notation.

Appendix 11. VSP LDEV Ownership and Performance

By Charles Lofton and Alan Benway

One of the most significant architectural differences between USP-V and VSP is the introduction of LDEV ownership by global processors (the MPU on each VSD board). Previous high-end RAID architectures (99xx, USP, USP-V) had no concept of LDEV ownership. Any FED processor (the CHP, or MP) could potentially handle host I/O (on the two ports it controlled) for any LDEV. Each MP processor had to acquire Shared Memory locks to ensure that LDEV updates were processed in the correct order, and to guarantee that a consistent view of the LDEV's data was presented to each host.

Over the years, while increased processor speeds allowed for much higher IOPS rates, the speed of access to metadata in the discrete global Shared Memory system (four boards) remained about the same. Therefore each new generation of processors on average was spending a greater portion of its time waiting for Shared Memory locks instead of getting useful work done. Thus Hitachi decided to adopt an LDEV ownership model for the VSP.

LDEV ownership means that a single MPU (the logical processor on each VSD board) is responsible for processing both host I/O and back end I/O for all of the LDEVs it owns. So, for example, when a host command comes in to read an 8KiB block from LDEV 01:00, the receiving FED DA ASIC processor would direct the request to the MPU that owns LDEV 01:00. Say that MPU-01 owns LDEV 01:00. MPU-01 would direct an I/O job to a BED board to read the 8KiB of data from disk, and then move that data into MPU-01's cache region. MPU-01 would then tell the originating FED processor (DA ASIC) to read the data from an address in MPU-01's cache region for transfer to the requesting host. No other MPU could perform this function for LDEV 01:00 (except in the event of VSD board failure, when the companion board of that installed feature would take on the load).

Performance Advantages

The performance advantage of LDEV ownership is that the MPU can store the metadata for the LDEVs it owns in its local memory on the VSD board. This means that all metadata reads come from local memory on the VSD, and are therefore much faster than the reads from the global Shared Memory system as required on our previous architectures.

LDEV ownership also significantly reduces metadata lock contention, since normally only one or two cores of the four cores on an MPU are working on a single LDEV at any given point in time, and no more than four cores will ever need the same lock. Metadata updates are write-through to global Control Memory system. LDEV ownership also means reduced contention for these metadata updates, since for any given LDEV, no more than four cores will need to update global shared memory.

In summary, LDEV ownership has the following performance advantages:

- Faster access to metadata
- Reduced lock contention
- Takes full advantage of modern, fast processors

Potential Performance Drawbacks

Unfortunately there are also a few pitfalls to be aware of with LDEV ownership. First and most obvious is that the CPU resources available for a given LDEV are limited to the four cores on the MPU that owns the LDEV. This can lead to imbalances in CPU utilization across VSD boards. We have seen cases in the field and test labs in which a single LDEV keeps all four cores on an MPU fully occupied, while other MPUs on different VSD boards sit idle. The MPU is the bottleneck in such cases. The most

straightforward way to resolve an MPU bottleneck is to stripe a logical volume across the MPU's physical volumes with a host volume manager. For example, instead of using a single 4TB LDEV, one could create 8 LDEVs of 512GB each. On an 8-VSD system, each of these LDEVs could be owned by a different MPU. One could then use a host volume manager to create a 4TB striped volume composed of these 8 LDEVs. I/O to such a striped volume would no longer be limited by the processing power of a single VSD.

The second thing to be aware of with LDEV ownership is default VSD assignment when creating LDEVs. The performance group has found that it's usually advantageous to have LDEV ownership in a Parity Group distributed across multiple VSDs. Distributing LDEV ownership is particularly important for Parity Groups composed of high-performance devices such as SSDs or FMDs. For example, a single RAID-5 (7D+1P) FMD Parity Group may be capable of 600,000 or more random read IOPS if all of this Parity Group's LDEVs were assigned to a single VSD, IOPS would be restricted to perhaps 125,000. As a general rule of thumb, evenly distribute LDEV ownership from a high-performance Parity Group across all available VSDs. When an array is initially configured, or when LDEVs are created with Storage Navigator, the default VSD assignment is round robin by PG.

For example, all LDEVs from PG 1-1 would be assigned to MPB0, LDEVs from PG 1-2 would be assigned to MPB1, and so on. The default (most common) assignment scheme leads to an ownership distribution in which a single VSD owns all the LDEVs from a PG. The default ownership distribution is generally acceptable for standard HDDs, but should be changed as noted above for high-performance PGs composed of SSDs or FMDs. Creating LDEVs in other scenarios may lead to a different VSD assignment scheme. For example, when creating LDEVs with a raidcom script, the VSD assignment may be pure round robin (each new LDEV goes on a different VSD, regardless of which PG it comes from). A pure round robin ownership distribution will generally perform well with all types of PGs. So, the VSP requires you to be aware of VSD assignment when creating LDEVs.

Cache Allocation per LDEV

LDEV ownership is also related to cache allocation. Each MPU initially gets an equal share of cache, but a MPU that owns a very busy set of LDEVs can borrow cache from other MPUs that have less work to do. But since cache reallocation is an overhead operation, performance will generally be better if LDEV ownership is balanced so that cache allocation among the MPUs remains stable.

DPVol Considerations

DPVOL ownership also has important performance implications. DPVOLs are a special type of "thin-provisioned" LDEV in which physical space is incrementally allocated in 42MB pages on a Dynamic Provisioning pool of disks. DPVOLs consume CPU core and cache resources on the MPU that owns them, just as standard LDEVs do. Therefore, DPVOL ownership should generally be distributed across multiple MPUs to balance load and to avoid single-VSD IOPS limits. The HDP disk pool from which DPVOLs are carved is composed of Pool Volumes. Pool Volumes are standard LDEVs that provide the physical storage space for the pool. Pool Volume ownership by MPU has not (so far) been found to have significant performance implications.

CLPR Usage Considerations

Finally, note that LDEV ownership is related to CLPR usage. A certain amount of cache is reserved for each CLPR for each MPU. This is so regardless of whether the MPU owns any resources in that CLPR. So for example, assume CLPR1 has been created to limit the amount of cache consumed by a set of DPVOLs, and assume DPVOL ownership is divided between MPU-01, MPU-02, and MPU-03. Even though MPU-00 owns none of the DPVOLs assigned to CLPR1, MPU-00 will always retain a certain amount of

CLPR1 cache. While some of MPU-00's CLPR1 cache could eventually be reallocated to other MPUs, the amount of CLPR1 cache retained by MPU-00 will be essentially wasted.

Summary

To summarize the potential drawbacks of LDEV ownership:

- Need to manually balance LDEV ownership among MPUs to ensure optimal use of CPU cores and cache.
- Must be aware of MPU assignment when creating new LDEVs. LDEV ownership in high-performance PGs should be distributed evenly across all VSDs.
- Need to consider DPVOL/LDEV ownership when allocating resources to CLPRs. An MPU that owns no CLPRx resources will waste a certain amount of CLPRx cache.



Corporate Headquarters 2845 Lafayette Street, Santa Clara, California 95050 USA
Contact Information: + 1 408 970 1000 www.hds.com / info@hds.com

Asia Pacific and Americas 2845 Lafayette Street, Santa Clara, California 95050 USA
Contact Information: + 1 408 970 1000 www.hds.com / info@hds.com

Europe Headquarters Sefton Park, Stoke Poges, Buckinghamshire SL2 4HD United Kingdom
Contact Information: + 44 (0) 1753 618000 www.hds.com / info.uk@hds.com

Hitachi is a registered trademark of Hitachi, Ltd., and/or its affiliates in the United States and other countries. Hitachi Data Systems is a registered trademark and service mark of Hitachi, Ltd. In the United States and other countries.

Microsoft is a registered trademark of Microsoft Corporation.

Hitachi Data Systems has achieved Microsoft Competency in Advanced Infrastructure Solutions.

All other trademarks, service marks, and company names are properties of their respective owners.

Notice: This document is for informational purposes only, and does not set forth any warranty, express or limited, concerning any equipment or service offered or to be offered by Hitachi Data Systems. This document describes some capabilities that are conditioned on a maintenance contract with Hitachi Data Systems being in effect, and that may be configuration-dependent, and features that may not be currently available. Contact your local Hitachi Data Systems sales office for information on feature and product availability.

Hitachi Data Systems sells and licenses its products subject to certain terms and conditions, including limited warranties. To see a copy of these terms and conditions prior to purchase or license, please go to <http://www.hds.com/corporate/legal/index.html> or call your local sales representatives to obtain a printed copy. If you purchase or license the product, you are deemed to have accepted the terms and conditions.

© Hitachi Data Systems Corporation 2013. All Rights Reserved.